

Università degli studi di Padova
Dipartimento di Scienze Statistiche

Corso di Laurea Magistrale in
Scienze Statistiche



Tesi di Laurea

**SORVEGLIANZA STATISTICA DI DATI VIDEO PER LA
RILEVAZIONE DI ANOMALIE TERMICHE NEI
PROCESSI DI PRODUZIONE ADDITIVA**

Relatrice: Prof.ssa Giovanna Capizzi
Dipartimento di Scienze Statistiche

Laureanda: Chiara Mattioli
Matricola n. 2037892

Anno Accademico 2022/2023

Indice

Introduzione	5
1 Opportunità e criticità nella produzione additiva	11
1.1 Introduzione alla produzione additiva	12
1.2 Processi <i>Powder Bed Fusion</i>	15
1.3 Tipologie di difetti di produzione	16
1.4 Rilevazione delle firme del processo	18
2 Rilevazione di anomalie termiche nei processi SLM	23
2.1 Il controllo statistico di processo nella produzione additiva . .	23
2.2 Sorveglianza di video: Stato dell'Arte	26
2.3 Manifestazione delle anomalie termiche nei video	29
3 La metodologia ST-PCA per la rilevazione di anomalie termiche	35
3.1 Obiettivi della metodologia ST-PCA	35
3.2 Struttura dei dati	36
3.3 Descrizione della metodologia	41
3.3.1 Calcolo delle ST-PCA e delle statistiche di controllo . .	41
3.3.2 Sistema di segnalazione basato sul metodo delle k medie	44
3.4 Osservazioni sui risultati ottenuti	46
4 Carte di controllo per la rilevazione di anomalie termiche	51
4.1 Idea e obiettivi della metodologia	51
4.2 Descrizione della metodologia	53
4.2.1 Simulazione dei video con il metodo bootstrap a blocchi	54

4.2.2	Calcolo dei limiti di controllo con il metodo di bisezione	57
4.3	Le statistiche di controllo utilizzate	59
4.3.1	Statistica EWMA	60
4.3.2	Statistica WCuSum	61
4.4	Implementazione della metodologia proposta	62
4.5	Analisi dei risultati: performance in controllo delle carte	65
5	Studio di simulazione	69
5.1	Metodo di simulazione delle anomalie termiche	69
5.1.1	Osservazioni sulla procedura di simulazione	73
5.2	Studio di simulazione	74
5.3	Sviluppi futuri	78
	Conclusioni	81
	A Elementi per l'implementazione della metodologia	85
	B Codice R implementato	89
B.1	Manipolazione delle strutture video	89
B.2	Implementazione del bootstrap a blocchi	91
B.3	Calcolo delle Statistiche di controllo	93
B.4	Calcolo dei limiti di controllo	101
B.5	Calcolo delle ARL_{IC} osservate	105
B.6	Simulazione di anomalie termiche	107
B.7	Calcolo delle ARL_{OC}	118
	Bibliografia	121

Introduzione

La produzione additiva (*additive manufacturing*) è sempre maggiormente utilizzata in ambito industriale. Questa tecnologia ha avuto una grossa evoluzione negli ultimi anni permettendo la realizzazione di prodotti con geometrie particolarmente complesse, che non sarebbero riproducibili con le tecniche più tradizionali. Questa tecnica di fabbricazione innovativa realizza i prodotti a partire da un modello digitale tridimensionale, aggiungendo gradualmente materiale strato dopo strato.

Tra le metodologie di produzione additiva maggiormente utilizzate vi sono i processi *Selective Laser Melting* (SLM), dove gli strati vengono realizzati dal passaggio di un laser che fonde selettivamente e solidifica un materiale in polvere. Questa procedura, che permette di ottenere oggetti veramente dettagliati e particolari, richiede però costi elevati, e per questo motivo prodotti difettosi devono essere evitati il più possibile.

Questo elaborato vuole proporre una metodologia di sorveglianza, basata su carte di controllo, per la rilevazione di anomalie termiche, note in letteratura con il termine di *hot spot*, nei processi SLM. Queste sono tra le principali cause di difetti geometrici nel prodotto finale e vengono causate da un errato trasferimento di calore tra il laser ed il materiale. Gli *hot spot* sono rilevati attraverso l'analisi di dati video che riprendono il processo di fusione di un singolo strato.

Nell'ultimo decennio sono stati fatti numerosi passi avanti nello sviluppo di metodologie in grado di sorvegliare questi processi. Tuttavia, a causa della difficoltà riscontrata nel delineare la dinamica del processo in condizioni di stabilità, non sono ancora stati implementati, per questo tipo di anomalia, dei sistemi di sorveglianza, online, dotati di opportuni limiti di controllo in

grado di segnalare tempestivamente l'insorgenza delle anomalie termiche. In particolare, partendo dal lavoro presentato da Colosimo e Grasso (2018), l'obiettivo di questa tesi sarà dunque quello di definire una procedura per il calcolo di limiti, che garantiscano un prefissato numero medio di falsi allarmi. Per quanto concerne l'organizzazione dell'elaborato, nel Capitolo 1 verrà brevemente descritta la tecnologia di produzione additiva, dando particolare risalto ai processi SLM e ai possibili difetti che possono insorgere durante la produzione.

Il Capitolo 2 presenterà lo stato dell'arte in cui si colloca questo lavoro, ovvero la sorveglianza statistica di dati video e la sua applicazione nella rilevazione di anomalie termiche.

La metodologia proposta da Colosimo e Grasso (2018), punto di partenza per la costruzione dei limiti di controllo, sarà argomento del Capitolo 3. Qui verranno innanzitutto definiti gli obiettivi e gli elementi di novità di tale metodo e saranno poi presentate le varie fasi della sua implementazione.

Nel Capitolo 4 sarà invece descritta una procedura per il calcolo dei limiti di controllo per la rilevazione delle anomalie termiche. In particolare, si mostreranno le difficoltà che caratterizzano tale procedura e le varie fasi necessarie alla sua implementazione. Si mostrerà come ottenere dei valori critici, i limiti, in grado di stabilire se il processo SLM, che si sta sorvegliando sia, o meno, in controllo, garantendo un certo numero di falsi allarmi.

Infine, nel Capitolo 5, sarà descritto un nuovo metodo di simulazione di anomalie termiche, e verrà sviluppato uno studio di simulazione per studiare il comportamento della metodologia proposta, quando applicata a video rappresentativi del processo fuori controllo.

Capitolo 1

Opportunità e criticità nella produzione additiva

L'obiettivo principale di questo elaborato è proporre una metodologia per la costruzione di carte di controllo, con l'obiettivo di rilevare particolari anomalie che possono insorgere in una tipologia di processi di fabbricazione innovativa, nota come produzione additiva (*additive manufacturing*). In particolare si vogliono rilevare delle anomalie termiche, chiamate in letteratura *hot spot*, spesso causa di difetti geometrici nei prodotti finali ottenuti.

A tale scopo, in questo primo capitolo introduttivo, verrà brevemente descritta la tecnica di produzione additiva, dando particolare risalto ad una determinata classe di processi, definiti *powder bed fusion*, su cui verteranno i capitoli successivi.

Verranno presentate le maggiori opportunità e discusse le principali criticità di questa tecnica di fabbricazione. In particolare si metteranno in evidenza le tipologie di difetti che possono manifestarsi durante la produzione, insieme agli strumenti e alle modalità utilizzati per rilevarli, sottolineando infine il livello di misurazione in cui si colloca il sistema di sorveglianza che verrà proposto in seguito.

1.1 Introduzione alla produzione additiva

L'*Additive Manufacturing* (AM) è una tecnologia innovativa che ha iniziato ad essere utilizzata durante gli anni '90 (si veda Gibson et al., 2021). Inizialmente era chiamata *Rapid Prototyping*, in quanto veniva principalmente impiegata nella rapida costruzione di prototipi o modelli che sarebbero poi stati messi in commercio.

Con la rapida evoluzione di questi macchinari, la qualità dei prodotti ottenuti ha cominciato ad aumentare in modo significativo. Inoltre la flessibilità delle materie prime utilizzabili, la possibilità di ottenere forme molto complesse e l'opportunità di realizzare un prodotto intero, ovvero non suddiviso in più parti assemblabili, hanno reso questa tecnologia quasi indispensabile in molti ambiti produttivi. Oggi la produzione additiva è impiegata da moltissime industrie ed il suo utilizzo è noto nel linguaggio comune con il termine: stampa tridimensionale (*3D printing*).

Esempi di utilizzo della produzione additiva si possono trovare nell'industria aerospaziale per la realizzazione di componenti leggeri e complessi, che riescono a ridurre il peso complessivo dell'aeromobile, migliorando così prestazione ed efficienza. Largo utilizzo delle stampe in 3D avviene anche in campo medico per produrre protesi personalizzate, modelli anatomici per la pianificazione chirurgica, attrezzature mediche specializzate, o persino tessuti e organi artificiali. Nella Figura 1.1 sono raffigurati alcuni elementi biomimetici di ambito medico realizzati attraverso la produzione additiva. Altri esempi si possono trovare anche nel settore energetico per la realizzazione di componenti per impianti di generazione di energia, oppure in ambiti creativi, quali oreficeria e design per la creazione di gioielli e prodotti unici, come mostrato nella Figura 1.2.

L'elemento che caratterizza la produzione additiva, e che ne permette la realizzazione di oggetti anche molto complessi, di cui quelli riportati nelle immagini nelle Figure 1.1 e 1.2 sono solo un esempio, consiste nella lavorazione del materiale strato su strato, partendo dall'elaborazione di un modello digitale 3D, specificato da dati CAD (*Computer Aided Design*). Il prodotto si ottiene quindi attraverso una continua aggiunta e lavorazione della materia prima in strati successivi, dove ognuno di questi è una sottile sezione



Figura 1.1: Esempi di prodotti biomimetici e personalizzati, realizzati tramite tecnologie di produzione additiva (Colosimo et al., 2018).

trasversale del prodotto derivato dai dati CAD originali. Questa tecnologia avanzata permette di ottenere un prodotto finale costituito da un unico elemento integrale. Tale caratteristica porta moltissimi vantaggi: consente di ottenere prodotti più flessibili e leggeri e consente un elevato risparmio di costi e tempo per l'assemblaggio delle parti.

Inoltre, grazie a questa tecnologia di costruzione a strati successivi, è possibile ridurre enormemente sprechi di materiale, rispetto alle metodologie tradizionali di fabbricazione. In queste ultime, infatti, viene utilizzato un approccio di sottrazione del materiale, ovvero partendo da un blocco o da una lastra della materia prima, le parti in eccesso vengono rimosse fino a ottenere la forma desiderata. I processi di sottrazione possono quindi generare una quantità significativa di scarti e detriti. Nella produzione additiva, invece, vengono utilizzate tecniche di deposizione mirate, in cui viene impiegata solo la quantità di materiale effettivamente necessaria per costruire l'oggetto strato dopo strato, evitando così l'accumulo di residui.

Ad oggi le macchine utilizzate nella produzione additiva differiscono tra loro per principalmente tre aspetti:



Figura 1.2: Esempi di prodotti realizzati con produzione additiva nell'industria energetica (in alto), e nei settori creativi (in basso) (Colosimo et al., 2018).

- i materiali che possono essere utilizzati per la realizzazione del prodotto;
- il modo in cui vengono definiti i singoli strati e
- come gli strati vengono poi legati l'uno all'altro a formare il prodotto finale.

Queste caratteristiche definiscono le specifiche proprietà dei processi, tra cui la velocità, la qualità e i costi dei prodotti ottenuti, e possono essere utilizzate per classificare le varie tipologie di processi AM. In Pham e Gault (1998), per esempio, viene proposta una categorizzazione a due dimensioni, basata sul metodo con cui vengono realizzati i singoli strati e sul tipo di materiale utilizzato. Negli ultimi anni questa categorizzazione è stata aggiornata da Gibson et al. (2021), dove vengono definiti 7 gruppi di processi AM.

In particolare questo elaborato si colloca nella categoria di processi di produzione additiva chiamata *Powder Bed Fusion* (PBF). Questa classe raggruppa tutti quei processi che generano il prodotto finale attraverso la lavorazione sequenziale di strati di polvere, i quali vengono solidificati utilizzando una

determinata fonte di energia, più comunemente un laser di scansione o un fascio di elettroni.

1.2 Processi *Powder Bed Fusion*

Come riportato in Gibson et al. (2021), i processi di produzione additiva PBF sono stati tra i primi ad essere sviluppati e, grazie alla loro grande versatilità, rimangono ancora tra i più utilizzati. Sono infatti adatti alla lavorazione di polimeri e metalli e, in misura minore, anche a quella di ceramiche e compositi.

Nei processi PBF gli oggetti vengono realizzati depositando e solidificando strati sottili di polvere di un certo materiale, uno sopra l'altro.

Tramite un sistema di deposizione di polvere, definito *recoating system* (si veda Figura 1.3), questa viene distribuita su una piattaforma di costruzione. Successivamente un raggio laser, o un fascio di elettroni, viene utilizzato per fondere selettivamente la polvere in determinate aree, secondo il disegno specificato dal modello CAD.

Il laser, o il fascio di elettroni, viene controllato da un sistema di scansione che guida la fusione della polvere strato dopo strato, sino al completamento del prodotto.

Dopo la fusione di ogni strato, la piattaforma si abbassa di una piccola quantità e un nuovo strato di polvere viene distribuito sopra quello precedente. Questo processo si ripete fino a quando l'intero prodotto viene ultimato (si veda Figura 1.3 per una illustrazione del processo PBF).

Come ampiamente discusso in Grasso e Colosimo (2017), attualmente i processi PBF si dividono principalmente in due gruppi. I primi sono definiti a fusione laser selettiva (SLM), dove la fonte di energia è un laser, mentre l'altro gruppo di processi utilizza come fonte di energia un fascio di elettroni, e vengono per questo chiamati processi PBF a fascio di elettroni (EBM).

In particolare, in questo elaborato, verranno trattate tecniche di sorveglianza statistiche applicate a processi SLM. Queste macchine utilizzano un letto di polveri metalliche, generalmente leghe di alluminio, acciaio, titanio o altre leghe, come materiale di partenza. Un laser ad alta energia viene indirizzato su specifiche aree del letto di polvere, fondendo e saldando insieme il mate-

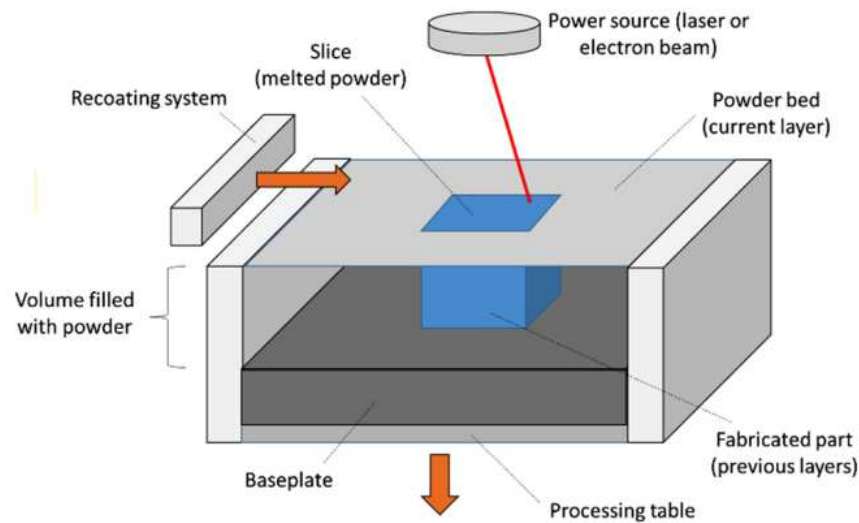


Figura 1.3: Schema di un generico processo PBF (Grasso e Colosimo, 2017).

riale in conformità con le istruzioni del modello CAD. Questo processo viene ripetuto per ogni strato fino a quando l'intero prodotto viene ultimato.

1.3 Tipologie di difetti di produzione

I processi AM sono molto articolati e sfruttano macchinari complessi, per questo motivo è facile che possano insorgere dei problemi durante le varie fasi della produzione, i quali potrebbero poi portare a prodotti finali imperfetti. Comprendere le tipologie di difetti e le loro cause è il primo passo per poter implementare delle tecniche di sorveglianza efficaci.

In questa sezione verranno trattati brevemente i principali difetti che possono interessare gli oggetti realizzati con le tecnologie PBF. Per una lettura più approfondita si faccia riferimento a Grasso e Colosimo (2017).

Un prodotto ottenuto con processi SLM o EBM può essere interessato da uno o più di questi difetti:

- **Porosità:** si tratta di vuoti di diversa forma e dimensione all'interno del materiale fuso, questi possono trovarsi sia all'interno del singolo strato, sia su strati adiacenti, oppure sulla superficie esterna alla parete (si vedano le immagini riportate nella Figura 1.4).

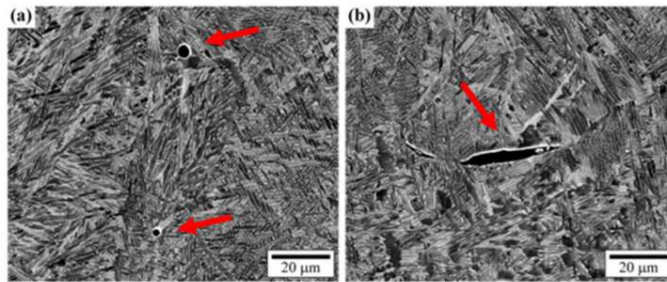


Figura 1.4: Esempio di due pori sferici (immagine a) e di un poro aciculare (immagine b) in un processo EBW (Grasso e Colosimo, 2017).

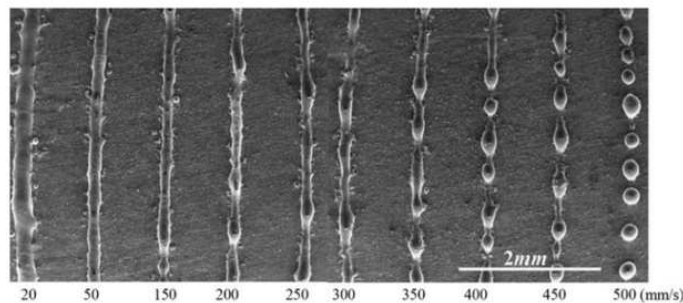


Figura 1.5: Esempio di pallonatura in un processo SLM (Grasso e Colosimo, 2017).

- **Pallonatura:** si verifica quando il materiale fuso si solidifica in sfere invece che in strati solidi, ciò costituisce un grande ostacolo alla connessione tra gli strati (si veda Figura 1.5).
- **Difetti geometrici:** le parti dei prodotti possono mostrare diversi tipi di deviazioni dimensionali o geometriche dal modello nominale, come deformazioni di varia natura, tra cui arricciamenti o bordi super rialzati. Questi difetti incidono fortemente sulla qualità e sulla stabilità del processo, in quanto peggiorano l'accuratezza del prodotto e, inoltre, data la tipologia di fabbricazione, possono propagarsi agli strati successivi e interferire con il sistema di deposizione, aumentandone l'usura e influenzando l'uniformità del letto di polvere (si vedano le immagini nella Figura 1.6).

In Grasso e Colosimo (2017) vengono mostrate anche altre tipologie di difetti, quali fessurazioni, disomogeneità microstrutturali e impurità del prodotto. Tutti questi errori di produzione possono avere cause molto differenti, legate per esempio all'attrezzatura utilizzata, alla qualità della polvere, oppure anche a particolari scelte di pianificazione della produzione. Possibili soluzioni per ridurre queste tipologie di difetti, prettamente legate ai macchinari e ai materiali utilizzati, sono: un maggiore controllo delle tecnologie adoperate e delle materie prime. Molto più difficilmente risolvibili, e rilevabili, sono invece quelle cause legate esclusivamente al processo, in particolare quelle associate all'interazione tra il fascio di luce, la polvere sciolta ed il materiale precedentemente fuso. Queste dinamiche danno origine alla maggior parte delle categorie di difetti sopra elencate, che rendono le parti prodotte imperfette. In molti casi anche azioni correttive sul prodotto finale non sono sufficienti, o addirittura applicabili, inoltre, l'introduzione di una fase di post-produzione per la correzione degli errori, richiederebbe costi ulteriori. È quindi essenziale sviluppare metodologie per la sorveglianza in tempo reale dei processi di produzione additiva, implementando tecniche di rilevamento volte a raccogliere informazioni sulla stabilità del processo, e sulla qualità del prodotto, mentre il pezzo viene realizzato strato dopo strato.

1.4 Rilevazione delle firme del processo

Le metodologie di sorveglianza, per verificare se il processo che si sta monitorando è ancora stabile, necessitano di informazioni in tempo reale (*in situ*). Durante le varie fasi della produzione possono essere rilevate delle specifiche quantità, definite in letteratura come firme (*signature*) del processo. In Grasso et al. (2021) vengono definiti 5 livelli di misurazioni *in situ*, in cui rilevare determinate *signature*, le quali sono informative del corretto iter di produzione (per una migliore comprensione si faccia riferimento alla Figura 1.3):

- **Livello 0:** comprende le quantità che possono essere misurate con sensori incorporati nei macchinari dei sistemi di AM. Le *signature* otte-

nute sono esplicative della corretta calibrazione e manutenzione degli strumenti.

- **Livello 1:** comprende le misurazioni raccolte una o più volte per ciascuno strato, prima o dopo la sua realizzazione. Sono per esempio quantità che misurano la disomogeneità del letto di polvere e le caratteristiche geometriche dello strato e della sua superficie.
- **Livello 2:** comprende le caratteristiche che possono essere misurate mentre il raggio laser è in azione, quindi mentre lo strato viene realizzato.
- **Livello 3:** comprende le misurazioni relative alla vasca di fusione, ovvero al recipiente che viene riempito con lo strato di polvere del materiale da stampare.
- **Livello 4:** comprende le *signature* che forniscono informazioni sui fenomeni che si verificano sotto lo strato corrente, come la profondità della vasca di fusione oppure la formazione di pori.

Per il corretto funzionamento dei processi SLM è dunque essenziale realizzare, per ogni livello, dei sistemi di sorveglianza efficaci.

Come anticipato all'inizio del capitolo, questo elaborato si pone l'obiettivo di implementare una metodologia di sorveglianza per la rilevazione in tempo reale di anomalie termiche. Queste anomalie, note in letteratura come *hot spot*, sono zone di surriscaldamento che possono verificarsi all'interno dei singoli strati e sono responsabili della maggior parte dei difetti geometrici rilevati nel prodotto finale (Figura 1.6). Vengono causate da un errato trasferimento di calore dal bagno di fusione al materiale circostante e solitamente si verificano nelle zone rivolte verso il basso, negli angoli acuti e nelle pareti sottili che sono maggiormente circondate da polvere sciolta. Le quantità necessarie per identificarle devono quindi essere rilevate mentre il laser di fusione è in azione, ovvero nel secondo livello di misurazione.

Le *signature* maggiormente utilizzate per la rilevazione di queste anomalie sono le mappe di calore, e i profili di raffreddamento. Questi vengono acquisiti dalla analisi di video che riprendono il processo di fusione del letto di

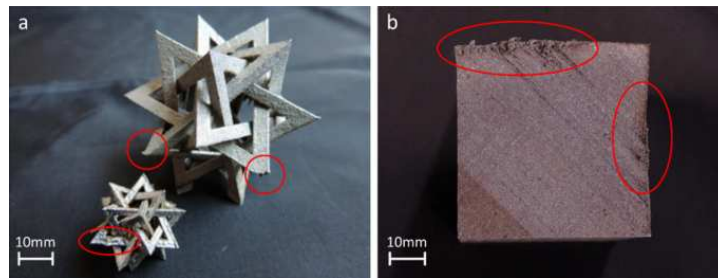


Figura 1.6: Esempi di difetti geometrici in prodotti realizzati con processi SLM: angoli acuti sporgenti (immagine **a**), lati sostenuti in modo errato (immagine **b**) (Grasso e Colosimo, 2017).

polvere nei singoli strati.

L'implementazione di un sistema di sorveglianza per la rilevazione di *hot spot* in processi SLM, si tradurrà quindi nella costruzione di una metodologia di sorveglianza statistica per dati strutturati (dati video). Questo argomento sarà affrontato nel Capitolo 2.

Capitolo 2

Rilevazione di anomalie termiche nei processi SLM

In questo capitolo verranno presentate le metodologie e gli strumenti per la sorveglianza statistica di processi SLM per la rilevazione di anomalie termiche, attraverso l'analisi e l'elaborazione di dati video. In particolare sarà fatta una panoramica sulle principali caratteristiche delle tecniche di controllo statistico, e delle maggiori difficoltà nella loro applicazione alla sorveglianza di processi di produzione additiva.

Verrà quindi illustrato lo stato dell'arte nella sorveglianza di immagini e video, sottolineando i maggiori ostacoli nell'implementazione di strumenti di sorveglianza per flussi di immagini tra loro correlate.

2.1 Il controllo statistico di processo nella produzione additiva

Come citato in Colosimo et al. (2018), la qualità della produzione e la ripetibilità dei processi sono tra le barriere più importanti da superare nella produzione additiva e, seppure siano stati fatti numerosi passi avanti nell'ultimo decennio, è ancora limitato l'apparato di metodologie strutturate per la sorveglianza della stabilità dei processi tipici produttivi.

Nel controllo statistico di processo (SPC) vengono utilizzate tecniche stati-

stiche per verificare e controllare le prestazioni di un processo produttivo, il quale ha una sua variabilità naturale ineliminabile, tuttavia col passare del tempo possono incorrere cause speciali che aumentano tale variabilità. Quando ciò avviene si dice che il processo non è più in condizioni stabili, o non è più in controllo (IC), ma è andato fuori controllo (OC) e quindi non offre più prestazioni desiderabili. Compito delle tecniche statistiche, che vengono raggruppate sotto il termine di SPC, è quindi quello di definire dei sistemi di sorveglianza che riescano a individuare il prima possibile allontanamenti del processo dalla situazione di stabilità. Per una lettura introduttiva all'SPC si rimanda a Qiu (2013).

La maggior parte delle tecniche di SPC si suddividono in carte di controllo di Fase 1 e di Fase 2. Con le prime l'obiettivo è quello di identificare un campione di osservazioni in controllo, con il quale poter descrivere il processo quando questo è stabile. Con le carte di Fase 2, invece, il processo viene sorvegliato in tempo reale, confrontando opportune statistiche, dette di controllo, che sintetizzano le informazioni sui nuovi campioni provenienti dal processo che si vuole sorvegliare, con dei limiti di controllo *ad hoc*, calcolati sulla base della dinamica in controllo descritta in Fase 1.

I limiti di controllo sono costanti critiche, che in base alla tipologia di cambiamento che si vuole rilevare possono essere unilaterali o bilaterali, variabili nel tempo o fissi. Questi consentono di determinare se il processo è in controllo, oppure no, ad ogni istante t in cui si sta sorvegliando il processo. Se la statistica di controllo, calcolata al tempo t , che chiamiamo Q_t , eccede i limiti di controllo, allora viene chiamato un allarme, con un certo margine di errore, in caso contrario il processo è considerato stabile. I limiti di controllo vengono calcolati fissando un numero medio di falsi allarmi.

Con il termine generale *Average Run Length* (ARL) si indica il numero medio di campioni estratti prima che la carta di controllo lanci un allarme. La *Run Length* (RL) è quindi il numero di campioni necessari a segnalare un allarme durante una singola traiettoria osservata. Sia Q_t una generica statistica calcolata sul campione osservato all'istante t , e siano (LCL,UCL) i limiti di controllo inferiore e superiore. La RL è così definita:

$$RL = \inf\{t > 0 : Q_t \notin [UCL, LCL]\} \quad (2.1)$$

I limiti della carta vengono quindi costruiti in modo da fornire un valore prefissato dell'ARL quando il processo è in controllo, chiamato ARL_{IC} .

$$ARL_{IC} = E_{IC}[RL] \quad (2.2)$$

Fissato un valore dell' ARL_{IC} , la carta di controllo di Fase 2 viene implementata cercando di ridurre il più possibile l' ARL_{OC} , ovvero il numero medio di campioni necessari a segnalare una situazione di fuori controllo, quando il processo non è stabile. In generale, sia τ l'istante in cui il processo va fuori controllo, si ha quindi:

$$ARL_{OC} = E_{OC}[RL - \tau + 1 | RL \geq \tau] \quad (2.3)$$

Si tenga presente che fissare un valore di ARL_{IC} equivale a fissare una certa probabilità di falso allarme: minore è il valore dell' ARL_{IC} , maggiore sarà tale probabilità. Questo valore influenza la reattività della carta quando applicata a processi fuori controllo, in quanto valori elevati di ARL_{IC} generalmente conducono a valori più elevati dell' ARL_{OC} . Stabilire un valore per l' ARL_{IC} sul quale definire i limiti di controllo è una scelta che deve essere presa dall'operatore considerando diversi aspetti, specifici del tipo di processo che si vuole sorvegliare. In alcune produzioni, infatti, l'aver un elevato numero di falsi allarmi può essere più dispendioso che non lasciarsi scappare alcuni veri allarmi.

In alcune contesti, i limiti possono essere calcolati in modo analitico, mentre nella maggior parte dei casi reali, in cui ciò non è possibile, i limiti vengono calcolati attraverso metodi di simulazione.

Nell'ambito della produzione additiva la complessità nell'implementazione di schemi di controllo con caratteristiche note in controllo, è dovuta principalmente alla difficoltà che si riscontra in Fase I nel descrivere il processo in controllo.

Infatti, nell'AM, i processi vengono riassunti da particolari *signature* ottenibili in tempo reale, le quali, nella maggior parte dei casi, sono dati complessi derivanti da immagini o video, su cui può essere molto complicato identificare un pattern in controllo. Ancora più complesse sono tutte quelle situazioni in cui non è proprio possibile ottenere un campione IC sufficientemente ampio, come nel caso di produzioni a *short run*, in cui gli oggetti vengono prodotti

in piccole quantità, oppure *one shot*, dove ogni prodotto è unico e viene realizzato singolarmente.

Queste difficoltà rendono la sorveglianza statistica in questo ambito un territorio ancora poco esplorato ed in continua evoluzione.

2.2 Sorveglianza di video: Stato dell'Arte

Le anomalie termiche sono identificabili attraverso una accurata analisi di dati video. Questi vengono acquisiti grazie all'utilizzo di specifiche videocamere, che riprendono il processo di fusione di uno strato di polvere dalla finestra dell'ambiente di costruzione (*build chamber*) (si veda Figura 2.1).

I video non sono altro che una sequenza di fotogrammi (*frames*), ovvero immagini, che riassumono le informazioni del processo, descrivono la sua dinamica, fotografandolo in istanti successivi. Il numero di fotogrammi di cui è composto un video è determinato dal numero di *frames* per secondo (fps) della telecamera utilizzata. Nei contesti di produzione additiva il numero di fotogrammi al secondo è generalmente molto elevato, e consente di cogliere più informazione possibile.

Negli ultimi anni, in SPC, sono stati fatti numerosi progressi nella sorveglianza statistica di dati complessi, e in particolare nella sorveglianza di immagini. Questo tipo di dato permette infatti di catturare molta informazione relativa ai processi sopra descritti e, per questo motivo, l'implementazione di metodologie di sorveglianza di immagini si è estesa a contesti molto differenti. Oltre a quello prettamente industriale si pensi all'ambito della sicurezza pubblica per la prevenzione di atti criminali, oppure al settore dei trasporti per monitorare i flussi di persone, e al monitoraggio ambientale per la sorveglianza di fenomeni estremi. Attualmente vi sono moltissime evoluzioni delle metodologie nell'SPC e il loro utilizzo si è ben ampliato dal semplice contesto industriale, sviluppando approcci sempre più moderni e versatili.

Questo sviluppo è stato possibile grazie all'evoluzione di tecnologie all'avanguardia e all'avanzamento di modelli di intelligenza artificiale (AI) di *computer vision*. Con questo termine si fa riferimento a quel campo dell'AI che si occupa di sviluppare algoritmi per l'acquisizione, l'elaborazione, l'analisi e la comprensione di informazioni visive provenienti da immagini o video (si

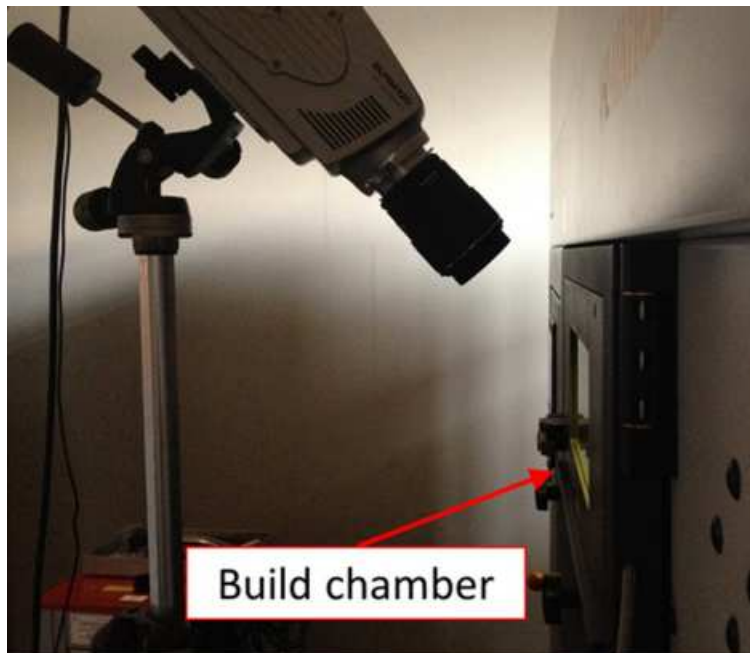


Figura 2.1: Acquisizione di un video di un processo di fusione (SLM) (Colosimo e Grasso, 2018).

vedano Russ e Neal, 2016 e Chen, 2015).

In Malik et al. (2023), per esempio, vengono utilizzate reti neurali per la sorveglianza di schemi spaziali, mentre in Dong e Catbas (2021) sono discusse le principali metodologie di sorveglianza di immagini basate su algoritmi AI per il monitoraggio della salute a livello sia locale che globale. In questi casi non sono state definite delle vere carte di controllo, ma dei sistemi di allarme basati su algoritmi di classificazione.

Molti autori hanno invece adattato le tecniche più tradizionali di SPC alla sorveglianza di dati immagine. In Armingol et al. (2003), per esempio, viene definita una metodologia per la sorveglianza dei singoli pixel, portando alla costruzione di una carta di controllo per ognuno di questi. Altri autori, invece, per evitare l'implementazione in parallelo di numerose carte di sorveglianza, hanno proposto la costruzione di opportuni indici sintetici delle immagini (si vedano Tong, Wang e Huang, 2005 e Park e Shrivastava, 2014).

Un altro approccio, basato sul metodo *Generalized Likelihood Ratio* (GLR), che utilizza le informazioni spaziali e temporali nelle immagini, è stato pro-

posto da Megahed et al. (2012) per rilevare sia la posizione del difetto, sia il momento in cui questo è avvenuto in un flusso di immagini. Tale approccio è stato poi esteso da He et al. (2016), per la rilevazione contemporanea di diverse tipologie di guasti. Elementi di novità sono stati poi presentati in Zuo, He e Zhang, 2020, con l'utilizzo di una carta di controllo *Exponentially Weighted Moving Average* (EWMA) (Qiu, 2013) per l'identificazione della dimensione e della posizione del difetto in un processo produttivo.

Vi è poi un'ampia gamma di sistemi di sorveglianza di immagini che utilizza metodi di riduzione della dimensionalità. In numerosi articoli sono state proposte estensioni spaziali o temporali della analisi delle componenti principali (PCA) (si veda Jolliffe, 2002) per la sintesi delle informazioni contenute nelle osservazioni, e la successiva applicazione di tecniche di SPC ai dati trasformati. Esempi di applicazioni si possono trovare in Lu, Plataniotis e Venetsanopoulos (2008), Qiu (2020), Chen e Wells (2022) e Eslami et al. (2023).

Tutte le metodologie sinora citate propongono tecniche di sorveglianza per dati immagine, ipotizzati come replicazioni aleatorie di un medesimo pattern in controllo. Pochi studi sono invece stati condotti sulla sorveglianza di video, ovvero flussi di immagini in contesti in cui questa ultima ipotesi non è soddisfatta. Tra questi, di particolare importanza per la stesura di questo elaborato, è la metodologia proposta da Colosimo e Grasso (2018), che consiste in un innovativo sistema di sorveglianza per dati video, basato su una regola di allarme per l'individuazione di *hot spot* che sfrutta conoscenze ingegneristiche sul processo. In Colosimo e Grasso (2018) viene inoltre proposta una estensione delle PCA in grado di catturare sia la correlazione spaziale presente all'interno dei singoli fotogrammi, sia quella temporale che si osserva tra fotogrammi successivi. Questa metodologia verrà descritta nel prossimo capitolo, e dai risultati ottenuti in Colosimo e Grasso (2018) sarà quindi proposta, nel Capitolo 4, una procedura per l'implementazione di carte di controllo, per la rilevazione di anomalie termiche.

Ulteriori approcci alla sorveglianza di video, con diretta applicazione ai processi SLM per la rilevazioni di anomalie termiche, sono stati sviluppati negli anni successivi. In Bugatti e Colosimo (2022), il problema di rilevazione di *hot spot* viene trasformato in un problema di classificazione attraverso l'im-

plementazione di algoritmi AI, quali reti neurali e *Support Vector Machines* (SVM) (Hastie e Friedman, 2001). In Yang et al. (2023) viene invece descritta una metodologia di sorveglianza statistica di tipo gerarchico, in grado di rilevare diverse tipologie di anomalie che possono insorgere in livelli del processo SLM differenti.

Tuttavia, per quanto la letteratura sia in rapida evoluzione, per quanto di nostra conoscenza, non è ancora stata implementata in questi contesti una carta di controllo, delimitata da opportuni limiti, per l'individuazione di anomalie termiche. Infatti, i metodi proposti, si basano principalmente su sistemi di allarme che non sfruttano limiti di controllo, ma piuttosto conoscenze a priori relative al fenomeno oggetto di studio, oppure algoritmi di classificazione. La caratterizzazione del processo IC può avere però diversi vantaggi, primo tra tutti quello di calcolare degli opportuni limiti di controllo in modo da garantire una certa probabilità di falso allarme.

Obiettivo di questo elaborato sarà quindi quello di caratterizzare il processo di fusione dei singoli strati nelle macchine SLM in situazioni stabili, costruendo uno strumento di sorveglianza, ovvero una carta di controllo, che riesca a garantire un prestabilito valore dell' ARL_{IC} . Tale studio sarà oggetto del Capitolo 4.

2.3 Manifestazione delle anomalie termiche nei video

Come anticipato nel Capitolo 1, nei processi SLM, gli *hot spot* sono tra le principali cause di deformazioni dei prodotti finali. Nella Figura 2.2 viene mostrato il medesimo oggetto di forma irregolare proposto nella Figura 1.6, in cui sono evidenziati alcuni difetti geometrici causati dall'insorgenza di anomalie termiche, durante la produzione di alcuni strati.

Un *hot spot* si verifica quando il laser, fondendo la polvere, lascia dietro di sé un punto particolarmente caldo, ciò viene causato da un errato trasferimento di calore dallo strumento al materiale, il cui raffreddamento nelle zone interessate dall'anomalia è molto più lento rispetto alle normali zone di fusione. Questo comportamento anomalo può essere rilevato analizzando i video ter-



Figura 2.2: Difetti geometrici locali causati dall'insorgenza di anomalie termiche (Colosimo e Grasso, 2018).

mici, registrati ad altissima velocità da camere ad alta risoluzione (Figura 2.1), che riprendono tutto il processo di fusione del letto di polvere.

In questa sezione si vuole mostrare come la dinamica in controllo del processo di fusione si discosti da quella fuori controllo, e quindi in che modo le anomalie termiche si manifestano nella sequenza di fotogrammi successivi.

Nelle Figure 2.3 e 2.4 sono rappresentati 5 fotogrammi di dimensione 126×136 pixel, estrapolati rispettivamente da due video, composti da 161 fotogrammi consecutivi ottenuti ad elevata velocità (150 fps), che descrivono gli istanti del processo di fusione di uno strato. Questi video sono organizzati in due *array* 3-dimensionali, dove le prime due dimensioni descrivono la lunghezza e l'altezza dei fotogrammi, in pixel, mentre la terza dimensione identifica il fotogramma nel video. I pixel assumono valori compresi nel range $(0, 255)$. Valori piccoli identificano pixel poco luminosi, al contrario, valori grandi pixel molto luminosi. La Figura 2.3 riporta i fotogrammi di indici $j = \{48, 54, 60, 66, 72\}$ in uno stato IC, mentre la Figura 2.4 mostra gli stessi fotogrammi quando insorge un *hot spot* di dimensioni molto grandi (rettangolo di 100 pixel) al 48-esimo fotogramma. Il video fuori controllo è stato ottenuto simulando un'anomalia sul video originale in controllo. Del metodo di simulazione di *hot spot* si discuterà approfonditamente nel Capitolo 5. In questo momento

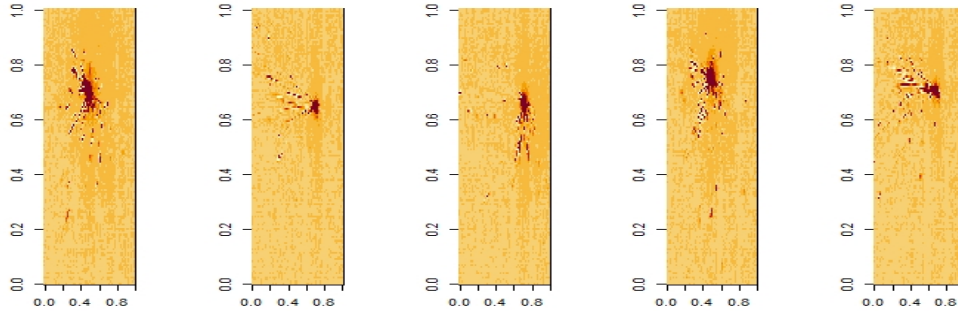


Figura 2.3: Processo in controllo: 5 fotogrammi consecutivi (indici: $j = \{48, 54, 60, 66, 72\}$) estrapolati dal video `dataIC` (Appendice A) che descrive un reale processo di fusione in assenza di *hot spot*.

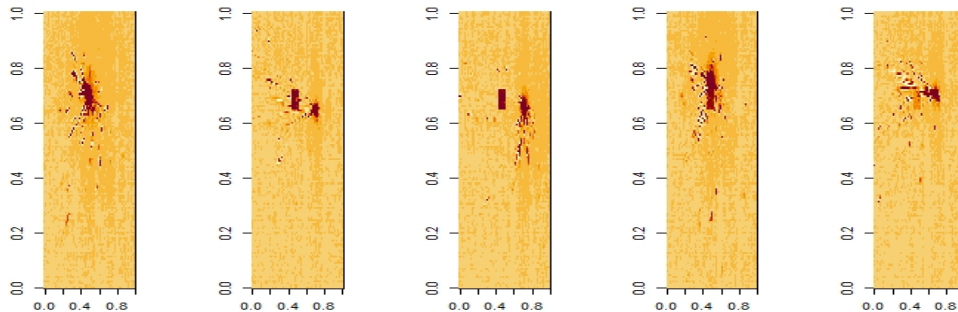


Figura 2.4: Processo fuori controllo: 5 fotogrammi consecutivi (indici: $j = \{48, 54, 60, 66, 72\}$) estrapolati dal video `dataIC` (Appendice A) che descrive un reale processo di fusione, sul quale è stato simulato un *hot spot* di dimensioni molto grandi (rettangolo di 100 pixel), di durata 25 fotogrammi, a partire dal 48-esimo fotogramma (utilizzando le Formule 5.2 e 5.3).

i fotogrammi sono rappresentati a solo scopo illustrativo, al fine di mostrare come si presenta il problema oggetto di studio. Si tenga presente che tra un fotogramma ed il successivo ve ne sono 5 che non sono stati rappresentati. I fotogrammi riportati riassumono quindi un intervallo di 25 istanti temporali in due processi analoghi, il primo in controllo, e il secondo fuori controllo.

Quando il processo è stabile (Figura 2.3) si riescono a distinguere due zone: quelle più scure sono le aree del letto di polvere illuminate dal laser, ovvero quelle in cui sta avvenendo la fusione della polvere, quelle immediatamente circostanti, oppure quelle relative a getti incandescenti. Le aree più chiare, invece, definite di *background*, raffigurano le zone del letto di polvere non illuminate dal laser. Mediamente si è osservato empiricamente che i pixel di *background* hanno valori compresi tra 70 e 100, mentre quelli che descrivono le aree colpite dal laser hanno luminosità massima, e quelli nelle zone circostanti raggiungono valori in media superiori a 200.

Quando il processo non è stabile (Figura 2.4), analizzando la dinamica temporale dei fotogrammi, è possibile individuare una terza area, ovvero quella relativa all'*hot spot*. L'anomalia termica si presenta come una zona di fusione che, invece di raffreddarsi velocemente e tornare alla luminosità di *background*, come avviene in situazioni di stabilità, rimane calda per molto tempo dopo il passaggio del laser. Dalle immagini nelle Figure 2.3 e 2.4 è possibile notare la differenza nei profili di raffreddamento tra i pixel che descrivono il processo di fusione IC, e quelli invece interessati dall'anomalia. Per poter cogliere queste dinamiche e sintetizzarle in opportuni strumenti di sorveglianza, è necessario tenere in considerazione la dipendenza temporale tra fotogrammi successivi. Data la natura del dato, le immagini di un singolo video, non sono rappresentative dello stesso istante produttivo, ma, come mostrato, colgono fasi successive del processo. Le immagini sono quindi correlate e nella sorveglianza di questi processi bisognerà tenere in considerazione di tale dinamica temporale. Nel prossimo capitolo verrà presentata la metodologia proposta da Colosimo e Grasso (2018), per la sorveglianza di video nella rilevazione delle anomalie termiche, i cui risultati saranno il punto di partenza per il disegno delle carte di controllo, obiettivo di questo elaborato, la cui descrizione sarà oggetto del Capitolo 4.

Capitolo 3

La metodologia ST-PCA per la rilevazione di anomalie termiche

In questo capitolo verrà descritta in modo dettagliato la metodologia per la rilevazione di anomalie termiche nei processi SLM, proposta da Colosimo e Grasso (2018). Saranno inizialmente presentati gli obiettivi e gli elementi di novità di questo strumento. Si descriveranno poi la struttura dei dati video utilizzata nelle analisi e le varie fasi dell'implementazione della metodologia, e saranno infine discussi alcuni risultati importanti. Si pone l'attenzione su questo innovativo sistema di sorveglianza in quanto sarà il punto di partenza per la costruzione di carte di controllo per la rilevazione di *hot spot*, la cui implementazione sarà oggetto del Capitolo 4.

3.1 Obiettivi della metodologia ST-PCA

La metodologia di sorveglianza basata sulla *Spatially Weighted T-mode* PCA (ST-PCA), proposta da Colosimo e Grasso (2018), porta un grande contributo nella sorveglianza di dati video, e in particolare nella rilevazione di anomalie termiche nei processi AM. La metodologia ST-PCA è in grado di cogliere la natura dinamica in controllo dei video; rilevando le anomalie termiche entro ciascuno strato della parte prodotta, sfruttando opportune conoscenze ingegneristiche. In particolare, attraverso una estensione spaziale

della *T-mode* PCA (introdotta da Jolliffe, 2002 e ripresa da Grasso et al., 2017), e l'introduzione di un sistema di allarme basato su un metodo di raggruppamento, è possibile controllare un processo di fabbricazione additiva, attraverso la sorveglianza *in situ* di fotogrammi acquisiti ad elevata velocità. Un altro elemento di novità di questo articolo, è l'uso della ST-PCA, per sintetizzare l'informazione delle *signature* dei video. Questa estensione della tradizionale analisi delle componenti principali permette di tenere conto sia della correlazione spaziale dei pixel interni ad un singolo fotogramma, sia di quella temporale tra fotogrammi successivi. Questo è possibile grazie all'elaborazione dei dati in una precisa struttura, e all'introduzione di una opportuna matrice di pesi nel calcolo della covarianza, di cui si parlerà rispettivamente nelle Sezioni 3.2 e 3.3.

3.2 Struttura dei dati

I video possono essere descritti in termini di un'array 3-dimensionale. Di seguito verranno definite le quantità per la loro elaborazione e la loro analisi. Sia $\mathcal{U} = \{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_j, \dots, \mathbf{U}_J\} \in \mathbb{R}^{M \times N \times J}$ l'array che descrive il video di un processo di fusione di un singolo strato, dove $M \times N$ è la dimensione, in pixel, di ciascun fotogramma, mentre J è il numero totale di fotogrammi che compongono il video, e saranno acquisiti in tempo reale durante la sorveglianza. La matrice $\mathbf{U}_j \in \mathbb{R}^{M \times N}$ descrive quindi il j -esimo fotogramma, con $j = 1, \dots, J$. Chiamiamo $u_{m,n} = u_d = \{u_{m,n,1}, \dots, u_{m,n,j}, \dots, u_{m,n,J}\}$ il vettore che caratterizza il profilo di intensità del d -esimo pixel, di posizione (m, n) , nei J fotogrammi, con $m = 1, \dots, M$, $n = 1, \dots, N$ e $d = 1, \dots, p$, dove $p = M \times N$ è il numero totale di pixel in un fotogramma.

La rappresentazione in formato *array* del flusso di immagini sarà definita *folded*, e lo schema relativo alla elaborazione online di questi dati è mostrato nella Figura 3.1, in cui è possibile osservare come, ad ogni istante, venga acquisito un nuovo fotogramma j . Questo verrà utilizzato per verificare se il processo è ancora in controllo, oppure se sono insorte delle anomalie termiche. L'applicazione di metodi di riduzione della dimensionalità, quale l'analisi delle componenti principali, che verrà di seguito applicata, richiede però la rappresentazione dei dati in un formato *unfolded*; con questo termine si in-

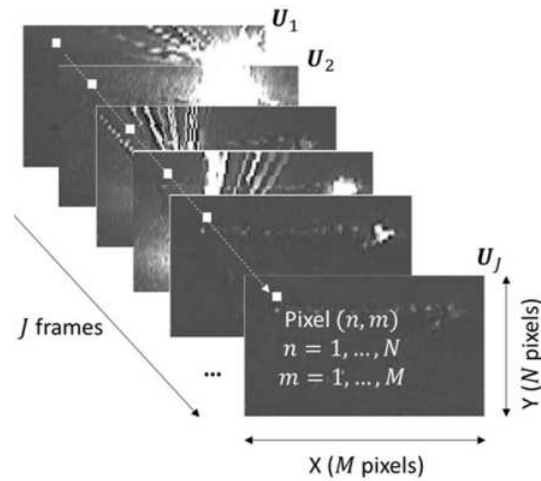


Figura 3.1: Rappresentazione schematica del flusso di fotogrammi (Colosimo e Grasso, 2018).

tende lo sviluppo dell'*array* \mathcal{U} in una opportuna matrice \mathbf{X} .

In generale le PCA (Jolliffe, 2002), e tutte le loro estensioni, sfruttano la decomposizione spettrale della matrice di varianza-covarianza dei dati, per trasformare un insieme di variabili tra loro correlate in un nuovo insieme di variabili non correlate, chiamate componenti principali. Queste sono una combinazione lineare delle variabili originali, e vengono ordinate in modo tale che la somma della varianza di ciascuna di esse sia una proporzione sufficientemente elevata della varianza dei dati originali. Questo permette, ad un numero limitato di componenti, di riassumere le informazioni contenute nei dati. Per l'applicazione di una analisi delle componenti principali è quindi indispensabile organizzare i dati in una opportuna matrice, definendo le unità statistiche e le variabili che vengono rilevate su queste.

In letteratura, per tenere conto della correlazione tra i pixel di un singolo fotogramma, oppure tra fotogrammi successivi, sono state introdotte differenti varianti delle PCA sui dati *unfolded*. In particolare, in Jolliffe (2002) sono state proposte due estensioni delle analisi delle componenti principali: la *S-mode* e la *T-mode*, le due procedure di *unfolding* dei dati, a partire dall'*array* \mathcal{U} , sono rappresentate rispettivamente a sinistra e a destra della Figura 3.2. Con la *S-mode*, \mathcal{U} viene convertita in una matrice $\mathbf{X} \in \mathbb{R}^{J \times p}$, con $p = M \times N$; ogni fotogramma viene trattato come una osservazione, e i singoli pixel assu-

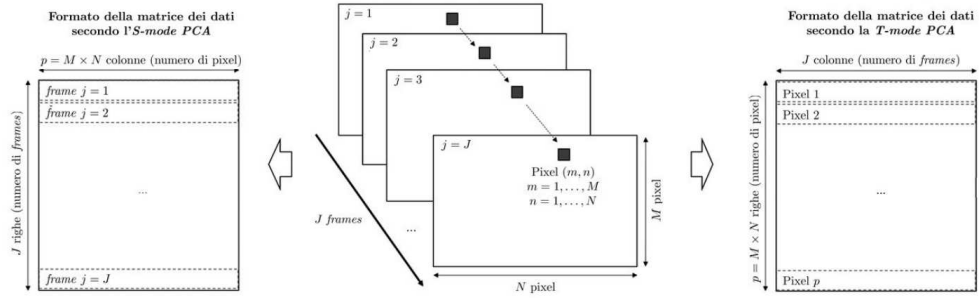


Figura 3.2: Procedure di *unfolding* dell'array originale del flusso di immagini nelle matrici di dati per la *S-mode* PCA (a sinistra) e la *T-mode* PCA (a destra) (Colosimo e Grasso, 2018).

mono il ruolo di variabili. La *S-mode* coglie quindi la correlazione tra i pixel, ovvero quella interna ad ogni fotogramma, senza considerare quella temporale tra fotogrammi successivi.

Al contrario, con la *T-mode*, \mathcal{U} viene trasformata in una matrice di dimensioni $\mathbf{X} \in \mathbb{R}^{p \times J}$, dove i fotogrammi sono trattati come variabili, mentre i singoli pixel come osservazioni. In questo caso ogni riga della matrice corrisponde ad un vettore $u_{m,n}$ e descrive il profilo termico del d -esimo pixel di coordinate (m, n) . Questo approccio consente di cogliere la autocorrelazione temporale dei pixel in fotogrammi successivi, perdendo però l'informazione spaziale tra pixel vicini.

Colosimo e Grasso (2018) propongono una versione spaziale dell'analisi delle componenti principali *T-mode*, la ST-PCA, in grado di cogliere sia la dipendenza temporale, sia quella spaziale attraverso l'introduzione di una opportuna matrice di pesi nel calcolo della matrice di varianza-covarianza. Questa versione estesa della PCA richiede la trattazione dei dati *unfolded* descritta con il metodo della *T-mode*, in cui i pixel sono le unità statistiche, e i fotogrammi sono le variabili.

La metodologia proposta da Colosimo e Grasso (2018) è stata implementata per poter sorvegliare un flusso di immagini in tempo reale. La matrice \mathbf{X} , che descrive l'intero video, non è quindi subito disponibile, ma ad ogni nuovo fotogramma viene sorvegliata la matrice $\mathbf{X}_j \in \mathbb{R}^{p \times j}$, la quale contiene le prime j colonne di \mathbf{X} , dove la j -esima variabile corrisponde all'ultimo fotogramma

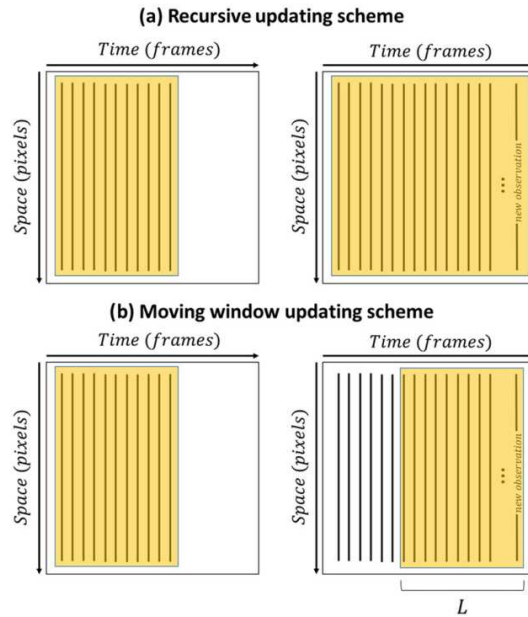


Figura 3.3: Rappresentazione schematica dei metodi di aggiornamento della matrice di dati *unfolded*: (a) metodo ricorsivo e (b) metodo a finestra mobile (Colosimo e Grasso, 2018).

registrato.

Da questo momento in poi, l'indice $j = 1, \dots, J$, fa quindi riferimento all'ultimo fotogramma considerato per la sorveglianza del processo, ed il relativo pedice j definisce le quantità ottenute utilizzando l'informazione disponibile fino al j -esimo fotogramma. Con J si indica invece l'ultimo fotogramma del video. L'aggiornamento della matrice, e la successiva fase di sorveglianza del processo, terminerà quando viene chiamato un allarme, oppure se il processo viene considerato in controllo durante tutta la registrazione del video, ovvero sino al J -esimo fotogramma finale.

È importante sottolineare che, ad ogni nuovo fotogramma, le unità statistiche della matrice di dati rimangono le stesse, mentre cambiano le variabili rilevate su di esse.

Sono possibili due tecniche di aggiornamento della matrice di dati in Colosimo e Grasso (2018), mostrate nella Figura 3.3. Con il metodo ricorsivo la matrice dei dati aumenta linearmente ad ogni nuovo fotogramma, passando da j a $j + 1$ variabili. In questo caso l'informazione storica viene conservata,

e ogni riga registra il profilo di raffreddamento del singolo pixel, dal primo al j -esimo fotogramma. Il metodo ricorsivo consente di sfruttare tutta l'informazione disponibile, tuttavia all'aumentare del numero di *frames* aumenta anche il costo computazionale richiesto dal metodo di sorveglianza. Il calcolo della ST-PCA, come tutte le altre metodologie di analisi delle componenti principali, è infatti basato sulla conoscenza della matrice di varianza-covarianza, la quale deve essere calcolata ad ogni nuovo fotogramma ottenuto. Sia \mathbf{S}_j la matrice di varianza-covarianza al j -esimo fotogramma. Utilizzando il metodo ricorsivo, una volta ottenuto un nuovo fotogramma, dovrà essere calcolata \mathbf{S}_{j+1} , la quale avrà dimensione $(j + 1) \times (j + 1)$, e richiederà il calcolo di $j + 1$ nuovi elementi rispetto alla matrice \mathbf{S}_j precedente. Si tenga comunque presente che gli elementi del quadrato superiore $j \times j$ di \mathbf{S}_{j+1} rimangono i medesimi della matrice \mathbf{S}_j . Questa informazione permette di calcolare, per ogni nuovo fotogramma, solo i nuovi $j + 1$ elementi, che descriveranno l'ultima riga e l'ultima colonna di \mathbf{S}_{j+1} , facendo risparmiare molto in termini di costo computazionale.

Con il metodo a finestra mobile, invece, vengono mantenuti solo gli ultimi L fotogrammi registrati, mentre i più vecchi vengono scartati. In questo caso il vettore $u_{m,n} = u_d = \{u_{m,n,(j-L+1)}, \dots, u_{m,n,j}\}$, relativo alla d -esima riga della matrice \mathbf{X}_j (e quindi al d -esimo pixel), non contiene tutta l'informazione storica, ma solo quella relativa agli L fotogrammi più recenti. Utilizzare il metodo a finestra mobile consente di lavorare con delle matrici di varianza-covarianza sempre della stessa dimensione $L \times L$, mantenendo quindi fisso il costo computazionale della metodologia di sorveglianza, rinunciando però a parte dell'informazione storica. Per ogni nuovo fotogramma possono essere calcolati solamente gli ultimi L elementi, relativi alle covarianze e alla varianza dell'ultima variabile. In questo caso la nuova matrice di varianza-covarianza \mathbf{S}_{j+1} si ottiene scartando la prima riga e la prima colonna da \mathbf{S}_j , elementi relativi alla variabile $j - L$, e aggiungendo, come ultime dimensioni, gli L elementi calcolati. Per quanto concerne la lunghezza L della finestra mobile, è importante osservare che valori troppo piccoli potrebbero non cogliere in modo appropriato la storia temporale dei pixel, mentre valori troppo grandi ridurrebbero i benefici computazionali che si ottengono rispetto al metodo ricorsivo.

A prescindere dal metodo di aggiornamento utilizzato è comunque importante sottolineare che, per poter essere avviata, la metodologia di sorveglianza necessita di un insieme (*batch*) di fotogrammi iniziali su cui calcolare la prima matrice di covarianza.

3.3 Descrizione della metodologia

In questa sezione saranno descritte le fasi principali della metodologia ST-PCA (Colosimo e Grasso, 2018). Inizialmente si presenterà il metodo di riduzione della dimensionalità utilizzato, e la successiva sintesi delle informazioni contenute nelle componenti principali attraverso l'utilizzo della T^2 di Hotelling. Si procederà poi alla descrizione del sistema di allarme introdotto per la rilevazione delle anomalie.

Tale procedura viene effettuata ad ogni nuovo fotogramma ottenuto. Quando un nuovo fotogramma è disponibile, la matrice dei dati viene aggiornata di conseguenza, con uno dei due metodi sopra descritti (Figura 3.3), e si ottiene quindi \mathbf{X}_j , sulla quale vengono calcolate le quantità necessarie alla sorveglianza.

3.3.1 Calcolo delle ST-PCA e delle statistiche di controllo

Introduzione della matrice di pesi \mathbf{W}

L'idea alla base della ST-PCA consiste nell'incorporare, nell'operatore di proiezione delle componenti principali, la correlazione spaziale tra i pixel, attraverso una matrice simmetrica $\mathbf{W} \in \mathbb{R}^{p \times p}$, dove $p = M \times N$ è il numero di pixel presenti in ogni immagine. Il generico elemento $w_{f,s}$ della matrice è un peso che quantifica la dipendenza spaziale tra l' f -esimo e l' s -esimo pixel, maggiore sarà il valore di $w_{f,s}$, maggiore sarà la loro dipendenza spaziale. In generale, pixel vicini sono tra loro maggiormente correlati; questa informazione viene utilizzata in molti operatori di lisciamiento delle immagini (si veda Russ e Neal, 2016). La matrice \mathbf{W} può in realtà essere vista come un particolare strumento di lisciamiento, ma invece di essere applicata alla matrice di pixel per modificare l'immagine, viene utilizzata per includere informazio-

ne spaziale nella correlazione tra i pixel. La matrice \mathbf{W} viene introdotta in quella di covarianza calcolata sui dati *unfolded* al j -esimo fotogramma (Figura 3.2, immagine a destra). Si ottiene così la seguente matrice \mathbf{S}_j che contiene l'informazione relativa sia alla correlazione spaziale sia a quella temporale.

$$\mathbf{S}_j = \frac{1}{p-1} (\mathbf{X}_j - \mathbf{1}_p \bar{X}_j^T)^T \mathbf{W} (\mathbf{X}_j - \mathbf{1}_p \bar{X}_j^T) \quad (3.1)$$

Nella (3.1), $\mathbf{X}_j \in \mathbb{R}^{p \times j}$ è la matrice di dati *unfolded* aggiornata al j -esimo fotogramma, $\bar{X}_j^T \in \mathbb{R}^{1 \times j}$ è il relativo vettore di medie campionarie e $\mathbf{1}_p$ è un vettore di $p \times 1$ elementi di valore uno.

Le seguenti 3 matrici sono state proposte da Colosimo e Grasso (2018):

$$\begin{cases} \mathbf{W}_1 : \{w_{f,s} = \frac{1}{d_{f,s}^2}\} \\ \mathbf{W}_2 : \{w_{f,s} = 1 \text{ se } d_{f,s} \leq r, w_{f,s} = 0 \text{ altrimenti}\} \\ \mathbf{W}_3 : \{w_{f,s} = (1 - \frac{d_{f,s}}{r^2}) \text{ se } d_{f,s} \leq r, w_{f,s} = 0 \text{ altrimenti}\} \end{cases} \quad (3.2)$$

Nella (3.2) $d_{f,s}$ rappresenta la distanza euclidea tra l' f -esimo e l' s -esimo pixel del fotogramma, mentre r è una soglia di riferimento, oltre la quale la dipendenza tra due pixel viene considerata non significativa. Gli elementi sulla diagonale sono posti pari a zero ($w_{f,f} = 0$) e i pesi delle matrici possono essere scalati per sommare a 1.

Calcolo delle componenti principali spaziotemporali (ST-PCA)

Una volta calcolata la matrice di covarianza \mathbf{S}_j , contenente l'informazione spaziale e temporale sui pixel, si procede con il calcolo delle ST-PCA.

La decomposizione spettrale di \mathbf{S}_j in componenti ortogonali, ha una forma analitica chiusa:

$$\mathbf{S}_j = \mathbf{E}_j \mathbf{\Lambda}_j \mathbf{E}_j'$$

dove $\mathbf{\Lambda}_j$ è una matrice quadrata $j \times j$ con gli autovalori di \mathbf{S}_j sulla diagonale principale, mentre \mathbf{E}_j è una matrice ortogonale le cui colonne sono gli autovettori associati agli autovalori di \mathbf{S}_j . Gli autovettori associano quindi un peso temporale ad ogni fotogramma disponibile sino a quel momento. La

matrice $\mathbf{Z}_j \in \mathbb{R}^{p \times j}$, delle componenti principali, si ottiene dal prodotto dei dati originali *unfolded* con la matrice di autovettori \mathbf{E}_j .

$$\mathbf{Z}_j = \mathbf{X}_j \mathbf{E}_j$$

Ogni colonna di \mathbf{Z}_j è una ST-PCA ed è una sintesi delle informazioni contenute nelle j variabili di \mathbf{X}_j , ovvero della storia dei pixel nei primi j fotogrammi. Come avviene nella analisi delle componenti principali standard, vengono selezionate le prime $G < j$ componenti più rilevanti, ovvero quelle che riescono a spiegare una certa percentuale desiderata della variabilità totale dei dati. Si ritiene importante ricordare che j è l'indice relativo all'ultimo fotogramma ottenuto nel corso della sorveglianza, e che quindi, le G componenti che si vogliono mantenere, hanno l'obiettivo di riassumere la storia temporale dei pixel fino al j -esimo fotogramma.

L'utilizzo delle componenti principali applicate a dati complessi, come possono essere le immagini, ha il duplice scopo non solo di ridurre la dimensionalità dei dati, ma soprattutto di ridurre il rumore che le osservazioni contengono. In particolare, in ambiti come la produzione additiva, in cui i video riprendono processi di fusione, e quindi processi complessi e particolarmente variabili, è essenziale riuscire a eliminare più rumore possibile, in modo da enfatizzare la vera componente di variabilità portata dal segnale.

Calcolo della statistica T^2 di Hotelling

Una statistica T^2 di Hotelling viene utilizzata per sintetizzare l'informazione storica di ciascun pixel in un unico indice, che possa essere utilizzato per verificare se il processo, al j -esimo fotogramma, è in controllo.

Nell'ambito della sorveglianza statistica tradizionale, la T^2 di Hotelling è una statistica multivariata, che misura la distanza del vettore delle medie dei dati dalla media attesa (si veda Qiu, 2013).

Nel nostro caso, la T^2 non viene utilizzata come statistica di controllo, ma solo come indice sintetico dei dati. Questa viene applicata alle prime G componenti principali che si desidera mantenere, le quali contengono la maggior parte del segnale e sono depurate dal rumore. Si ottiene così un vettore $T_j^2 \in \mathbb{R}^{p \times 1}$, di lunghezza p , contenente gli indici sintetici della storia dei p pixel fino al j -esimo fotogramma registrato.

Ogni elemento di T_j^2 , che chiamiamo $T_{j[d]}^2$, con $d = 1, \dots, p$, riassume quindi, in un unico valore, tutta la storia del d -esimo pixel, di coordinate (m, n) sino al j -esimo fotogramma, e viene calcolato nel seguente modo:

$$T_j^2(m, n) = T_{j[d]}^2 = \sum_{g=1}^G \frac{\mathbf{Z}_{j(d,g)}^2}{\lambda_g}, \quad d = 1, \dots, p \quad (3.3)$$

Nella (3.3) λ_g è la varianza della g -esima componente, con $g = 1, \dots, G$, mentre $\mathbf{Z}_{j(d,g)}$ è l'elemento, di posizione (d, g) , della matrice delle componenti principali, ed è quindi relativo al d -esimo pixel della g -esima componente.

L'idea alla base dell'utilizzo di questa particolare sintesi dei dati è che aree delle immagini caratterizzate da differenti schemi di autocorrelazione temporale avranno anche valori differenti di T^2 .

Questa informazione permette di localizzare andamenti anomali attraverso lo studio del vettore T_j^2 , per mezzo di una opportuna regola di allarme che sfrutta determinate conoscenze ingegneristiche sui processi SLM, e che verrà descritta di seguito.

3.3.2 Sistema di segnalazione basato sul metodo delle k medie

La regola di allarme descritta da Colosimo e Grasso (2018) rileva anomalie termiche attraverso lo studio del vettore delle statistiche $T_{j[d]}^2$ (3.3).

Viene proposto uno schema di sorveglianza che utilizza congiuntamente il metodo di raggruppamento delle k medie, per individuare dei gruppi nel vettore T_j^2 , e le conoscenze ingegneristiche legate al processo SLM (per una lettura approfondita sull'algoritmo delle k medie si veda Hastie e Friedman, 2001).

In Colosimo e Grasso (2018), l'algoritmo viene applicato alle statistiche $T_{j[d]}^2$, sotto l'ipotesi che, quando il processo è stabile, ovvero quando non sono presenti anomalie termiche, ci si aspetta la presenza di 2 soli gruppi: uno relativo ai pixel di *background*, e l'altro a quelli delle normali zone di fusione. Come discusso nella Sezione 2.3, in presenza di *hot spot*, invece, è plausibile la presenza di un terzo gruppo, relativo ai pixel che rappresentano le zone di fusione con un profilo di raffreddamento molto più lento rispetto a quello che ci si aspetterebbe in assenza di anomalie.

Sfruttando questa informazione, verrà quindi chiamato un allarme se, nella

distribuzione delle statistiche $T_{j[d]}^2$, è verosimile la presenza di un terzo gruppo. In questo caso il processo viene fermato, per verificare se si è trattato di un vero allarme. In caso contrario, il processo viene considerato in controllo e si procederà quindi con l'acquisizione di un nuovo fotogramma, e con l'aggiornamento della matrice di dati con uno dei due metodi riportati nella Figura 3.2.

Per determinare il numero ottimale k_{opt} di gruppi in cui partizionare il vettore T_j^2 , Colosimo e Grasso (2018) hanno proposto di selezionare k_{opt} cercando il gomito nella distribuzione delle devianze residue totali interne ai gruppi. La devianza residua totale, $SSW_j(K)$, è data dalla somma delle singole devianze interne ai K cluster identificati.

$$SSW_j(K) = \frac{1}{K} \sum_{k=1}^K SSW_{j,k} = \frac{1}{K} \sum_{k=1}^K \sum_{d \in C_k} (T_{j[d]}^2 - \bar{T}_{j,C_k}^2)^2 \quad (3.4)$$

Nella Formula 3.4, \bar{T}_{j,C_k}^2 rappresenta il valore medio delle statistiche $T_{j[d]}^2$ raggruppate nel k -esimo cluster, C_k . Il singolo elemento $SSW_{j,k}$, quantifica quindi l'eterogeneità interna al k -esimo gruppo. Più elevato è il valore di $SSW_{j,k}$, più le osservazioni interne a C_k , e quindi le statistiche $T_{j[d]}^2$, con $d \in C_k$, sono differenti tra loro.

All'aumentare del numero di gruppi, l'eterogeneità interna diminuisce, di conseguenza $SSW_j(K)$ è una funzione decrescente in K . Si tenga presente che identificando un numero di gruppi esattamente pari al numero di osservazioni, la devianza totale residua sarebbe appunto nulla.

Sapendo che solo due gruppi sono sufficienti per caratterizzare le osservazioni in controllo, l'indice $SSW_j(K)$ è diviso per il valore K al fine di penalizzare la selezione di troppi gruppi. La presenza di un gomito nella funzione identifica k_{opt} , ovvero il numero di cluster tale per cui una ulteriore partizione dei dati non diminuirebbe significativamente la devianza residua interna ai gruppi. Se viene individuato un punto di gomito in $K = 2$, allora il processo è considerato in controllo, in caso contrario è possibile che siano insorte delle anomalie termiche durante il processo di fusione.

Si osservi che $SSW_j(2)$, ovvero la devianza residua totale ottenuta partizionando i pixel in due cluster, può rappresentare essa stessa una statistica di controllo. Valori elevati di $SSW_j(2)$ suggeriscono infatti grande eterogeneità

tà all'interno dei gruppi, riconducibile alla presenza di anomalie termiche. Il prossimo capitolo sarà infatti incentrato sullo sviluppo di una procedura per l'implementazione di carte di controllo con le quali sorvegliare il processo, attraverso le $SSW_j(2, t)$. In particolare, la $SSW_j(2, t)$ sarà considerata l'osservazione, rappresentativa del funzionamento del processo, all'istante t . Con l'indice t ci si riferisce all'istante in cui diventa disponibile l'ultimo fotogramma j , con il quale viene calcolata $SSW_j(2, t)$. Nel seguito si scriverà più semplicemente $SSW_j(2)$, dal momento che vi è una relazione univoca tra l'istante t , in cui viene rilevata l'osservazione, e l'ultimo fotogramma j disponibile in quel momento. La differenziazione tra i due indici è infatti determinata dal *batch* di fotogrammi necessari per inizializzare la procedura.

3.4 Osservazioni sui risultati ottenuti

In questa sezione saranno brevemente riportati e commentati alcuni risultati di performance ottenuti negli studi di simulazione di Colosimo e Grasso (2018). Tali risultati saranno il punto di partenza per l'implementazione della procedura, per la costruzione di carte di controllo, che verrà descritta nel prossimo capitolo. Per maggiori dettagli sulle analisi sviluppate e sugli studi sperimentali eseguiti si rimanda all'articolo sopra citato.

Il video impiegato per queste analisi è un video reale, registrato a 150 fps ed è composto da 174 fotogrammi totali, alcuni dei quali sono stati riportati nella Figura 2.3 a scopo illustrativo. Il video in oggetto è stato ottenuto durante la realizzazione di uno strato ed è rappresentativo del processo SLM in controllo, e sarà utilizzato per l'implementazione della metodologia descritta nel prossimo capitolo. Su questo video, Colosimo e Grasso (2018), simulano alcune anomalie termiche di diversa durata e dimensione. Del metodo di simulazione impiegato da Colosimo e Grasso (2018), e di come si è deciso di modificarlo, per ottenere una simulazione più realistica, si discuterà nella Sezione 5.1.

Colosimo e Grasso (2018) hanno confrontato la metodologia di sorveglianza basata su ST-PCA con due metodologie alternative: la PCA multilineare (MPCA) (Lu, Plataniotis e Venetsanopoulos, 2008) e la *T-mode* (Jolliffe, 2002), considerando due percentuali di varianza spiegata dalle componenti

principali (50% e 80%), e utilizzando entrambi i metodi di aggiornamento della matrice di dati (Figura 3.3), con $L = 50$ per la finestra mobile.

La *T-mode* viene implementata con la stessa metodologia descritta per la ST-PCA, ma senza introdurre la matrice di pesi \mathbf{W} nel calcolo della covarianza, mentre la MPCA estende la *S-mode* (Jolliffe, 2002) al caso multivariato, attraverso l'utilizzo di una analisi delle componenti principali tensoriale (si rimanda a Lu, Plataniotis e Venetsanopoulos, 2008 e Yan, Paynabar e Shi, 2014). A differenza della *T-mode* e della ST-PCA, che associano un valore T^2 ad ogni pixel, la MPCA associa un valore T^2 ad ogni fotogramma. La MPCA non si è infatti rivelata robusta nella sorveglianza di flussi di immagini, portando sempre alla segnalazione di allarmi nonostante il video fosse in controllo, per ogni configurazione di parametri. Ciò non si è osservato con le altre due metodologie. Questo è causato dal fatto che la MPCA non sintetizza la storia temporale dei pixel sino al j -esimo fotogramma rilevato, ma riassume invece, in ogni statistica, la distribuzione dei pixel osservati nel j -esimo fotogramma di riferimento, senza considerare ciò che è avvenuto nei fotogrammi precedenti. Non essendo quindi in grado di cogliere la dinamica del processo, che si evolve fotogramma dopo fotogramma, la MPCA porta alla segnalazione di allarmi anche quando il processo è stabile.

Colosimo e Grasso (2018) hanno poi mostrato come l'introduzione delle dipendenze spaziali tra i pixel, attraverso l'utilizzo delle matrici di pesi \mathbf{W} , renda la ST-PCA estremamente più efficace della *T-mode* nel rilevare le anomalie termiche. Tale risultato rivela come, in questi contesti, è indispensabile tenere in considerazione sia la dipendenza temporale tra fotogrammi successivi, sia quella spaziale tra pixel vicini. L'introduzione di \mathbf{W} ha infatti conseguenze rilevanti sul numero di componenti necessarie per garantire una certa percentuale di varianza spiegata. L'implementazione della *T-mode* necessita di un grande numero di componenti principali, il quale cresce inoltre linearmente, con l'indice j , quando si utilizza il metodo di aggiornamento ricorsivo. Questo fa sì che il segnale relativo all'anomalia, quando presente, venga perso. La ST-PCA è invece robusta, ovvero il numero di componenti necessarie si mantiene costante durante tutta la traiettoria di sorveglianza, e richiede in generale un ridotto numero di componenti. Questo permette di rilevare più facilmente le anomalie termiche, dato che il relativo segnale non

viene mascherato dal rumore contenuto nell'elevato numero di componenti considerate.

La grande efficienza delle ST-PCA, nel riassumere la storia dei pixel, è il motivo per cui la metodologia proposta in questo elaborato, e che sarà discussa nel prossimo capitolo, utilizzerà questa tipologia di analisi delle componenti principali.

La ST-PCA verrà implementata utilizzando una specifica combinazione di parametri. In particolare, Colosimo e Grasso (2018) hanno mostrato come la scelta di una matrice di pesi \mathbf{W} , rispetto alle altre, non influenza in modo determinante la reattività della metodologia di sorveglianza. Per questo motivo, siccome il valore della soglia r , per il calcolo delle matrici \mathbf{W}_2 e \mathbf{W}_3 , deve essere selezionato a priori, sarà utilizzata \mathbf{W}_1 per il calcolo di \mathbf{S}_j nell'implementazione delle carte di controllo.

Si è inoltre deciso di utilizzare come soglia di varianza spiegata quella del 50%, in quanto Colosimo e Grasso (2018) mostrano come quella all'80% riduca la reattività della ST-PCA, e come metodo di aggiornamento, quello ricorsivo. Seppure la differenza non sia significativa, i risultati riportati da Colosimo e Grasso (2018) rivelano come questo metodo di aggiornamento renda la sorveglianza più reattiva nella rilevazione di anomalie di grandi dimensioni, con un ARL_{OC} circa pari a 31 fotogrammi, rispetto al metodo a finestra mobile, in cui l' ARL_{OC} ottenuto è circa pari a 34. Si fa presente che l'insorgenza di *hot spot* di piccole dimensioni e di breve durata difficilmente potrà avere un impatto significativo sulla qualità del prodotto finale, per questo motivo è molto più importante riuscire ad individuare *hot spot* di grandi dimensioni e di lunga durata, i quali possono effettivamente essere causa di difetti geometrici.

Si fa infine presente che Colosimo e Grasso (2018) hanno mostrato come inizializzare la procedura con 40 fotogrammi sia sufficiente per giungere a buoni risultati, di conseguenza, anche la metodologia proposta, utilizzerà un *batch* iniziale di 40 fotogrammi.

Capitolo 4

Carte di controllo per la rilevazione di anomalie termiche

In questo capitolo verrà proposta una metodologia per la costruzione di carte di controllo per la rilevazione di anomalie termiche in processi SLM. Questa metodologia utilizza un bootstrap a blocchi per la simulazione di un campione di video in controllo di Fase 1, e un metodo di bisezione, per il calcolo di opportuni limiti critici. Verrà innanzitutto illustrata l'idea alla base di questo strumento, presentandone i principali obiettivi. Saranno quindi descritte le fasi dell'implementazione, e discussi i vantaggi e i punti critici nell'utilizzo di questa metodologia di sorveglianza, e la sua applicazione attraverso tre diverse statistiche di controllo.

4.1 Idea e obiettivi della metodologia

Il vantaggio nell'utilizzare una carta di controllo, rispetto a una generica regola di allarme, come quella proposta da Colosimo e Grasso (2018), è la possibilità di controllare il numero medio atteso di falsi allarmi. Una carta di controllo permette infatti, sulla base di un limite critico, di segnalare quando il processo è fuori controllo, consentendo di prendere decisioni consapevoli relativamente all'interruzione della produzione.

Dati i costi computazionali associati all'implementazione di questa metodo-

logia, le carte di controllo verranno costruite per una sola combinazione dei parametri, descritti nella Sezione 3.4. In particolare, si è deciso di:

- utilizzare \mathbf{W}_1 (3.2) come matrice di pesi;
- mantenere un numero di componenti in grado di garantire il 50% di varianza spiegata;
- utilizzare come metodo di aggiornamento dei dati il metodo ricorsivo;
- inizializzare la sorveglianza con un *batch* di 40 fotogrammi iniziali. Si fa notare che anomalie termiche insorte all’inizio del processo possono quindi essere rilevati solo dopo il 40-esimo fotogramma.

Da questo momento in poi quindi, a meno che non venga esplicitato altrimenti, si farà riferimento ai due sistemi di sorveglianza (sistema di allarme presentato in Colosimo e Grasso (2018), e metodologia di seguito proposta per la costruzione di carte di controllo) con questa configurazione di parametri.

Come discusso nel precedente capitolo, la statistica $SSW_j(2)$ può essere rappresentativa del corretto funzionamento del processo di fusione. In particolare, valori elevati sono indicativi di una forte eterogeneità all’interno dei 2 gruppi individuati, e quindi della possibile presenza di pixel la cui storia temporale non è conforme a quella ipotizzata in condizioni di stabilità. Invece di sorvegliare il processo attraverso la regola di allarme introdotta da Colosimo e Grasso (2018), l’insorgenza di una anomalia termica sarà segnalata quando il valore della statistica osservata all’istante t , relativo al j -esimo fotogramma, eccede il limite di controllo.

É importante sottolineare la differenza e il legame tra i due indici j e t che saranno utilizzati: l’indice $j = 1, \dots, J$ si riferirà sempre all’ultimo fotogramma considerato per il calcolo della statistica. L’indice t , invece, si riferirà all’istante in cui viene calcolata la statistica. La differenza tra j e t è determinata dal *batch* di fotogrammi necessario per inizializzare la procedura. Dato j , si ha: $t = j - 40 + 1$. Per evitare complicazioni nella notazione, nell’elaborato verrà sempre utilizzato il pedice j , ad eccezione di quando ci saranno riferimenti alla RL (2.1), e all’ARL_{IC} (2.2).

Come verrà spiegato in seguito, la metodologia proposta sarà implementata per la sorveglianza del processo attraverso tre statistiche di controllo, trasformazioni di $SSW_j(2)$. Infatti, carte di controllo basate su statistiche differenti, sono sensibili in modo differente a diverse tipologie di cambiamento.

La carta $SSW(2)$, che utilizza come statistica di controllo l'osservazione $SSW_j(2)$ è una carta di tipo Shewhart (si veda Qiu, 2013), ovvero una carta che sorveglia il processo ad ogni istante, considerando la sola osservazione ottenuta in quel momento, ed è per questo sensibile a grandi cambiamenti, che sono quelli a cui noi siamo maggiormente interessati. Le carte di tipo EWMA (si veda Qiu, 2013) e WCuSum (si veda Shu, Jiang e Tsui, 2008), che saranno descritte nella Sezione 4.3, e applicate a questo studio, sono invece carte note in letteratura per essere sensibili a piccoli cambiamenti.

Con l'obiettivo di definire carte di controllo per la rilevazione di anomalie termiche nei processi SLM, si introdurrà una metodologia per calcolare i limiti di controllo, garantendo certi valori dell' ARL_{IC} (2.2). Si faranno inoltre presenti le principali criticità di questa metodologia, relativamente alla robustezza dei risultati ottenuti, e i motivi che le comportano. Si crede comunque che la metodologia che sarà qui descritta, possa essere utile per la sorveglianza di dati video, per diversi ambiti applicativi.

4.2 Descrizione della metodologia

Una breve introduzione alla SPC è stata fatta nella Sezione 2.1, in cui si è spiegato i limiti di controllo siano calcolati in modo tale da garantire un prefissato numero medio di falsi allarmi. Esistono situazioni standard in cui tali limiti possono essere calcolati analiticamente, sfruttando le conoscenze teoriche legate alle distribuzioni note delle statistiche di controllo, ma in molti casi reali, come quello in oggetto, risulta necessario ricorrere a metodi di simulazione. Tali metodi consentono di riprodurre il comportamento delle statistiche quando il processo è in controllo, e di calcolare quindi i limiti in modo da garantire una performance in controllo desiderata.

La metodologia che verrà implementata in questo elaborato utilizza un metodo di bisezione per il calcolo dei valori critici delle diverse statistiche di controllo. Non si utilizzerà, però, la versione classica del metodo di bisezio-

ne, la quale richiede che i campioni, su cui calcolare le relative statistiche e l'ARL_{IC}, vengano generati uno per volta durante la procedura. Sarebbe infatti troppo oneroso computazionalmente il calcolo di limiti di controllo per statistiche differenti, descritte in Sezione 4.3, che sono trasformazioni delle statistiche $SSW_j(2)$, il cui calcolo ha un costo computazionale a sua volta molto elevato. Si è perciò deciso di procedere per due fasi successive: inizialmente, verranno generati 500 video in controllo attraverso l'utilizzo di un bootstrap a blocchi. Successivamente si calcoleranno le varie statistiche in controllo, che verranno ricampionate con il metodo di bisezione, per trovare il valore critico h che garantisce un desiderato valore dell'ARL_{IC}. Ovviamente, la simulazione di un unico campione di 500 video in controllo, ha lo svantaggio di portare a risultati meno robusti rispetto a quelli che si otterrebbero con il metodo di bisezione tradizionale, in cui ad ogni iterazione dell'algoritmo si dovrebbe generare un nuovo campione IC. Tuttavia la metodologia descritta ha il grande vantaggio di proporre una procedura estremamente veloce e applicabile in quei contesti in cui il calcolo delle statistiche di controllo è computazionalmente oneroso.

Di seguito saranno descritti innanzitutto il metodo del bootstrap a blocchi, per la simulazione dei video, e successivamente il metodo di bisezione, per il calcolo del valore critico h , e infine saranno mostrati i risultati ottenuti in termini di performance in controllo delle carte.

4.2.1 Simulazione dei video con il metodo bootstrap a blocchi

Come discusso nel Capitolo 2, uno dei principali ostacoli nel disegno di una carta di controllo, nell'ambito della sorveglianza di processi AM, è avere a disposizione un campione in controllo che sia rappresentativo del processo che si vuole studiare. In questi casi possono essere utilizzati metodi di simulazione e/o ricampionamento. Tra questi, il bootstrap è uno dei più utilizzati (si veda Efron e Tibshirani, 1994). L'idea alla base del bootstrap è quella di ottenere un numero I di campioni simulati, estraendo con reinserimento le osservazioni a disposizione a partire dall'unico campione osservato, imitando così il processo di campionamento casuale alla base dell'inferenza frequentista.

Nel nostro caso il campione a disposizione è formato da un solo video rappresentativo del processo di fusione in controllo. Il video in oggetto, introdotto nel Capitolo 2, è composto da 174 fotogrammi di dimensione 126×136 pixel, ottenuti ad una frequenza di 150 fps. Dato che il primo e gli ultimi 13 fotogrammi fotografano il processo nei momenti iniziale e conclusivo, in cui il laser di fusione è spento, si è deciso di eliminarli. Includendo questi *frames*, si rischierebbe infatti di avere video simulati nei quali il laser rimane spento per un periodo molto lungo, cosa che non avviene nel processo in controllo. Il video sul quale si lavorerà è quindi composto da $J = 161$ fotogrammi totali. L'idea, in questo caso, è quella di utilizzare una estensione del bootstrap, un bootstrap a blocchi, per ottenere, a partire dal video originale, I nuovi video indipendenti tra di loro e che siano rappresentativi della dinamica del processo in controllo (Bühlmann, 2002). Il bootstrap a blocchi considera blocchi di dati, anziché singole osservazioni, come unità di campionamento, e viene utilizzato quando i dati presentano una struttura di dipendenza temporale, per preservare la correlazione entro i blocchi. Come descritto in Sezione 2.2, i video non sono altro che una sequenza di fotogrammi che descrivono la dinamica del processo in istanti successivi. Considerando quindi i *frames* come unità, è possibile simulare I nuovi video, ricampionando a blocchi i fotogrammi dal video originale. La qualità dei video simulati, in termini di rappresentatività del processo in controllo, dipende essenzialmente dalla lunghezza b (numero di fotogrammi sequenziali) dei blocchi. Il valore di b deve essere scelto in modo tale che la dipendenza temporale, osservata nel campione di fotogrammi originale, venga conservata nei video simulati.

In Politis e White (2004) (e in Patton, Politis e White, 2009) vengono presentate, in un unico quadro unificato, le diverse metodologie di bootstrap a blocchi per dati autocorrelati, fornendo importanti risultati asintotici sulla loro efficienza. In particolare vengono qui proposti alcuni utili metodi di selezione automatica per il valore ottimale di b , che chiameremo b_{opt} , e per la sua stima. Ricampionare blocchi di osservazioni di lunghezza b_{opt} permette di ottenere simulazioni in cui viene conservata la maggior parte della correlazione del campione originale.

Per il caso studio in oggetto verrà utilizzato il bootstrap a blocchi circolare (CB), che è il più tradizionale tra i metodi bootstrap per dati dipendenti (si

veda Bühlmann, 2002) e, grazie alla procedura proposta in Politis e White (2004), è possibile ottenere stime accurate di b_{opt} . Con il metodo CB, le osservazioni vengono considerate ordinate in una forma circolare. Più semplicemente, si ipotizzi di avere a disposizione un campione formato da x osservazioni sequenziali, e di volerne simulare un altro utilizzando blocchi di b osservazioni successive. Se il blocco del quale si sta ultimando la sequenza incontra la x -esima osservazione, viene utilizzata la prima come unità successiva, riprendendo così il conteggio dalle osservazioni iniziali.

I metodi di selezione del valore di b_{opt} , proposti da Politis e White (2004), sono però stati sviluppati per identificare la lunghezza ottimale dei blocchi bootstrap, quando le singole osservazioni sono quantità univariate. Nel nostro caso le osservazioni sono fotogrammi, ovvero unità complesse composte da $p = 126 \times 136 = 17136$ pixel. L'idea è quindi quella di considerare i pixel come unità elementari e i p profili delle loro intensità $u_{m,n}$, rilevati nel corso dei J totali fotogrammi, come p campioni autocorrelati. Per ogni pixel viene quindi calcolata la lunghezza ottimale dei blocchi bootstrap utilizzando il metodo proposto in Politis e White (2004).

Definiamo tali quantità $b_{opt,d}$, con $d = 1, \dots, p$. Il valore di b_{opt} viene scelto in modo tale che, nella maggior parte dei p profili, i blocchi di lunghezza b_{opt} siano sufficienti a conservare la correlazione esistente. Nella Figura 4.1 viene rappresentata, per ognuno dei $p = 17136$ pixel, la lunghezza ottimale $b_{opt,d}$ dei blocchi bootstrap. Empiricamente si è osservato come, utilizzando una lunghezza b_{opt} pari a 20, si riesca a conservare la autocorrelazione dei profili in più del 98% dei p campioni. Si può appunto vedere nella Figura 4.1 come la maggior parte dei valori $b_{opt,d}$ sia inferiore a 20, e solo un piccolo numero richieda valori superiori ed elevati, pari circa al doppio di b_{opt} . Dato che i pixel sono tra loro spazialmente dipendenti, una percentuale del 98% è ritenuta più che sufficiente per conservare la correlazione osservata nella sequenza di fotogrammi originale, in quanto si crede che la dipendenza tra i singoli campioni, ovvero tra i singoli pixel, sia in grado di compensare la correlazione temporale non conservata nel 2% dei profili.

In particolare, l'utilizzo di blocchi di 20 fotogrammi permette di simulare un grande numero di video differenti. Ogni video simulato deve essere formato da 161 fotogrammi, e viene quindi definito da 9 blocchi consecutivi. I bloc-

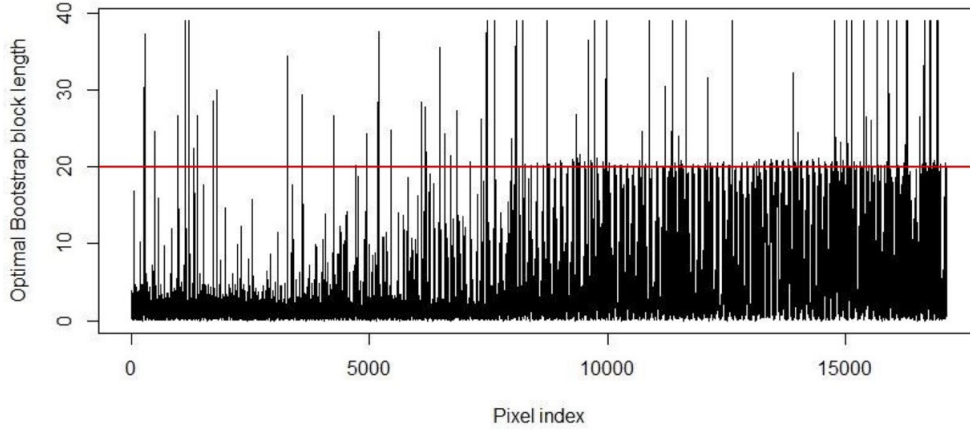


Figura 4.1: Lunghezza ottimale $b_{opt,d}$ dei blocchi bootstrap per ogni pixel, evidenziando il valore $b_{opt} = 20$.

chi sono univocamente determinati dal loro fotogramma iniziale, che viene campionato tra i 161 disponibili. Ciò significa che possono essere simulati $161^9 \simeq 7.2 \cdot 10^{19}$ video differenti.

Per il calcolo di opportuni limiti critici è necessario avere a disposizione un campione sufficientemente numeroso di video che descrivono il processo in condizioni di stabilità. Con il bootstrap a blocchi circolare sono quindi stati simulati $I = 500$ video rappresentativi del processo in controllo.

4.2.2 Calcolo dei limiti di controllo con il metodo di bisezione

Una volta simulati, attraverso il bootstrap a blocchi, i video che descrivono il processo in condizioni di stabilità, è possibile ricorrere a metodi numerici per il calcolo dei limiti di controllo. Dato il fenomeno oggetto di studio, è di interesse calcolare il solo limite superiore. Infatti, solo aumenti di $SSW(2)$ sono indicativi di profili di raffreddamento anomali. A questo scopo il valore critico h , oltre il quale sarà chiamato allarme, verrà calcolato attraverso il metodo di bisezione (si veda Qiu, 2013). Si tratta di un metodo iterativo che, partendo da un certo valore iniziale del limite, si avvicina, per step successivi, al valore critico h che soddisfa l' ARL_{IC} desiderato.

Per semplicità espositiva, il processo iterativo per il calcolo del valore critico, sarà quindi descritto per la sorveglianza di una generica statistica Q . Le statistiche Q vengono calcolate per ogni video in controllo $i = 1, \dots, I$, con $I = 500$, utilizzando la configurazione di parametri descritta in Sezione 4.1. Ogni video è composto da 161 fotogrammi. Dato il *batch* di 40 fotogrammi necessario per inizializzare la procedura per il calcolo delle ST-PCA, per ogni video i saranno quindi ottenute 122 statistiche Q in controllo: partendo dal 40-esimo fotogramma verrà calcolata una statistica $Q_{i,j}$ ad ogni j -esimo fotogramma successivo, con $j = 40, \dots, 161$, e $\forall i = 1, \dots, 500$.

Per quanto discusso all'inizio del capitolo, si fa notare che ad ogni fotogramma j , con $j = 40, \dots, 161$, è quindi associato un istante $t = j - 40 + 1$, per cui $Q_{i,j} = Q_{i,t}$, dove $t = 1, \dots, 122$. Con $Q_{i,t}$ indichiamo quindi la t -esima statistica, calcolata al t -esimo istante, per l' i -esimo video.

Per la ricerca del limite è innanzitutto necessario calcolare le statistiche Q per ogni video, e selezionare dei valori opportuni per le seguenti quantità: (i) A_0 : target dell' ARL_{IC} che si vuole garantire, (ii) $[h_L, h_U]$: estremi inferiore e superiore iniziali entro cui cercare h , (iii) ρ : accuratezza della stima dell' ARL_{IC} , e (iv) M : numero massimo di iterazioni. La ricerca del valore h si interrompe se l'algoritmo giunge a convergenza, oppure se si raggiunge il numero massimo di iterazioni.

Il limite h viene quindi trovato procedendo iterativamente per i seguenti step:

1. Calcolo di $h = (h_L + h_U)/2$
2. Calcolo della RL (2.1), numero di statistiche prima di chiamare un allarme, per ogni campione Q_i :

$$RL_i = \min\{t : Q_{i,t} > h\}$$

3. Calcolo dell' ARL_{IC} (2.2), numero medio di statistiche per chiamare allarme:

$$ARL_{IC} = \frac{1}{500} \sum_{i=1}^{500} RL_i$$

4. Se $ARL_{IC} \in [A_0 - \rho, A_0 + \rho]$ allora l'algoritmo si interrompe, perché giunto a convergenza, in caso contrario i valori di h_L e h_U vengono

aggiornati come segue e gli step precedenti vengono ripetuti:

$$\begin{cases} h_L = h_L; & h_U = (h_L + h_U)/2 & \text{se } ARL_{IC} > A_0 \\ h_L = (h_L + h_U)/2; & h_U = h_U & \text{se } ARL_{IC} < A_0 \end{cases} \quad (4.1)$$

Nel prossimo capitolo, per ogni statistica di controllo, saranno condotti studi di sensibilità relativamente al target A_0 dell' ARL_{IC} che si vuole garantire.

Come anticipato all'inizio del capitolo, solo aumenti delle statistiche sono rilevanti per l'individuazione di anomalie termiche. In particolare si fa notare che le statistiche $SSW(2)$ sono indici di variabilità, quindi assumono sempre valori positivi; lo stesso sarà verificato anche per le trasformate che saranno utilizzate. Da ciò deriva che il limite h sarà sempre contenuto tra 0 e il valore massimo delle $Q_{i,t}$; per questo motivo i valori di h_L e h_U saranno inizializzati a $[0, \max(Q_{i,t})]$ per ogni statistica Q . I valori dei parametri M e ρ sono invece stati posti pari a 20 e 0.0001 rispettivamente. Si è infatti osservato come con questi valori l'algoritmo sia sempre riuscito a convergere, impiegando mediamente dalle 13 alle 16 iterazioni.

Nella prossima sezione verranno presentate le carte che saranno implementate con questa metodologia, per la sorveglianza del processo di fusione SLM.

4.3 Le statistiche di controllo utilizzate

Nel caso studio in oggetto, $SSW_j(2)$ rappresenta sia l'osservazione, sia la statistica di controllo di tipo Shewhart rilevata al j -esimo fotogramma.

Come precedentemente discusso, in condizioni di stabilità, i pixel hanno intensità riconducibili a due possibili scenari: (i) pixel di *background* e (ii) pixel illuminati dal laser. Durante il processo, lo scenario in cui si collocano i pixel cambia in modo continuo, questo porta al disegno di profili molto dinamici e differenti tra loro, in cui è difficile identificare un pattern. La statistica $SSW_j(2)$, infatti, non sintetizza direttamente la dinamica dei pixel, ma l'eterogeneità rilevata tra queste dinamiche. Sotto l'ipotesi che in condizioni di stabilità 2 *cluster* siano sufficienti a spiegare il processo, $SSW_j(2)$, suddivide i pixel in 2 gruppi, e individua quanto i loro profili di intensità, appartenenti ad uno stesso gruppo, siano tra loro omogenei. Ciò significa che parte dell'informazione contenuta in $SSW_j(2)$ sarà anche contenuta in $SSW_{j+1}(2)$ e

così via. Le statistiche $SSW(2)$ sono quindi intrinsecamente correlate; questa loro caratteristica potrebbe rendere la statistica Shewhart inadatta, in quanto tendenzialmente utilizzata per sorvegliare processi le cui osservazioni, in istanti successivi, sono tra loro indipendenti. Per questo motivo può risultare interessante applicare alle osservazioni $SSW_j(2)$ opportune trasformate, che tengano conto di questa dipendenza. Con questo obiettivo vengono proposte tre statistiche di controllo: (i) $SSW(2)$, (ii) EWMA e (iii) WCuSum, dove (ii) e (iii) sono applicate a $SSW(2)$.

I corrispondenti limiti di controllo saranno individuati attraverso il metodo di bisezione, utilizzando come quantità Q una delle tre statistiche sopra citate. Una volta definito un limite per una certa quantità, può dunque essere attuato uno studio di sorveglianza su un nuovo video: ad ogni nuovo fotogramma registrato, viene calcolata una statistica di controllo, per la quale è stato calcolato il limite di controllo, e viene verificato se questa eccede o meno tale valore critico. In caso affermativo sarà chiamato un allarme.

Di seguito si descrivono le statistiche Q di tipo EWMA e WCuSum.

4.3.1 Statistica EWMA

Le carte di controllo *Exponentially Weighted Moving Average* (EWMA) e *Cumulative Sum* (CuSum) sono tra le carte con memoria più utilizzate (si veda Qiu, 2013). Il termine "con memoria" si riferisce al fatto che, a differenza di altre statistiche, queste cumulano l'informazione nel tempo. In particolare la EWMA conserva le informazioni storiche attraverso l'utilizzo di una media mobile pesata.

L'idea è quella di cumulare nel tempo le osservazioni $SSW(2)$. Queste ultime, come sopra specificato, contengono già l'informazione storica dei pixel, tuttavia è importante osservare che tale informazione viene sintetizzata in modo differente da fotogramma a fotogramma. L'informazione che il d -esimo profilo, relativo al d -esimo pixel, apporta alla statistica $SSW_j(2)$ è diversa da quella che apporterà alla statistica $SSW_{j+1}(2)$, ovvero lo stesso intervallo temporale, di un medesimo profilo, non viene egualmente riassunto nelle statistiche $SSW_j(2)$ ed $SSW_{j+1}(2)$. Utilizzando una statistica di tipo EWMA è possibile includere, nella statistica calcolata al $(j+1)$ -esimo fotogramma, l'in-

formazione su come il profilo era sintetizzato sino al fotogramma precedente. La statistica EWMA, al j -esimo fotogramma, sarà quindi così ottenuta:

$$E_j = \lambda SSW_j(2) + (1 - \lambda)E_{j-1} \quad (4.2)$$

Nella (4.2), $SSW_j(2)$ è l'osservazione rilevata al j -esimo fotogramma, mentre E_{j-1} è la statistica calcolata al fotogramma precedente. Il parametro $\lambda \in (0, 1)$ definisce la "memoria" della carta e deve essere scelto a priori. Maggiore è il valore di λ , più la carta dà peso alle nuove osservazioni ed è sensibile a cambiamenti grandi.

La carta EWMA, per essere implementata, necessita di un valore iniziale, il quale sarà posto pari alla statistica $SSW_j(2)$, con $j = 40$, ovvero alla prima statistica $SSW(2)$ calcolata.

4.3.2 Statistica WCuSum

La statistica WCuSum, introdotta da Shu, Jiang e Tsui (2008) è una estensione della CuSum tradizionale (Qiu, 2013), ed è stata sviluppata per rilevare piccole deviazioni da una media variabile nel tempo. Per una descrizione dettagliata si rimanda all'articolo sopra citato. In generale, il metodo proposto vuole stimare la media dinamica del processo in controllo attraverso l'utilizzo di una EWMA, e utilizzare tali stime per pesare l'incremento della media delle osservazioni durante la sorveglianza con una carta CuSum tradizionale. La statistica WCuSum viene calcolata, ad ogni nuovo fotogramma j osservato, nel seguente modo:

$$W_j = \max\{0, W_{j-1} + (e_j - \kappa) \cdot |\phi_j|\} \quad (4.3)$$

Nella (4.3) il parametro κ determina la sensibilità delle carte CuSum rispetto a deviazioni grandi o piccole dalla media. Valori di κ piccoli sono utili per l'individuazione di *shift* piccoli, al contrario valori grandi per l'individuazione di *shift* grandi (si veda Qiu, 2013). Le e_j sono i residui di cui si vuole sorvegliare l'incremento, calcolati come deviazioni delle osservazioni dalle stime ottenute adattando un modello autoregressivo. È stato precedentemente osservato come le $SSW(2)$ siano tra loro correlate; con la WCuSum, a differenza di quanto viene fatto con la EWMA, l'idea è quella di depurare le statistiche dalla correlazione che le caratterizza e sorvegliare i residui, che si crede

essere rappresentativi della dinamica in controllo della media. Studiando il correlogramma dei residui calcolati sui video in controllo, si è verificato come l'adattamento di un modello AR(1) alle osservazioni SSW(2) sia sufficiente a spiegare tutta la correlazione.

Ad ogni nuovo fotogramma, si calcola quindi la stima $\widehat{SSW}_j(2)$, regredendo l'ultima statistica $SSW_j(2)$ su quelle precedenti, e si ottiene di conseguenza il residuo e_j che entrerà nella Formula 4.3. Il primo residuo, relativo alla prima osservazione, sarà posto pari a 0.

L'elemento ϕ_j è invece il peso che viene associato al j -esimo incremento osservato. Si fa notare che, nel caso in cui $\phi_j = 1 \forall j$, si ottiene la carta CuSum tradizionale. I pesi ϕ_j vengono calcolati per mezzo di una statistica EWMA (Formula 4.4), che, in questo caso, non viene utilizzata come sintesi del processo, ma come stimatore della media dinamica dei residui, i quali, se il video che si sta sorvegliando è in controllo, sono tra loro non correlati.

$$\phi_j = \lambda e_j + (1 - \lambda)\phi_{j-1} \quad (4.4)$$

Ad ogni nuovo fotogramma la statistica WCuSum pesa il residuo osservato con un valore, che descrive la dinamica della media osservata sino al j -esimo fotogramma.

Nella seguente sezione viene descritta l'implementazione del sistema di sorveglianza proposto in questo capitolo.

4.4 Implementazione della metodologia proposta

In questa sezione verranno brevemente descritti i passaggi per l'implementazione della metodologia proposta. Tutti gli elementi utilizzati sono elencati nella Appendice A, mentre le funzioni, e i comandi R essenziali per il calcolo delle varie quantità, sono riportati in Appendice B.

Innanzitutto, vengono generati i 500 video in controllo. Per prima cosa, si calcolano le p lunghezze ottimali dei blocchi bootstrap per ciascun profilo ($b_{opt,d}$, con $d = 1, \dots, p$) e, in accordo con quanto discusso nella Sezione 3.3.1, viene trovato il valore b_{opt} . Si generano quindi i 500 video in controllo, a partire dal video originale, attraverso un bootstrap circolare di blocchi di lunghezza $b_{opt} = 20$. Le funzioni e i comandi utilizzati sono descritti in Appendice B.2.

In Appendice B.3 sono invece riportate le funzioni e i comandi per il calcolo delle statistiche. In particolare, viene innanzitutto calcolata la matrice \mathbf{W}_1 (3.2), indispensabile per ricavare le ST-PCA. Si fa notare che viene utilizzata la versione non scalata della matrice, in quanto è stato osservato che il suo utilizzo permette di ottenere statistiche di valori moderati, generalmente comprese nel range (0,1), senza modificarne la struttura o i risultati.

È stata quindi implementata tutta la metodologia per il calcolo, in ogni video, delle 122 statistiche $SSW(2)$. Innanzitutto, vengono calcolate le ST-PCA, seguendo i passaggi descritti in Sezione 3.3.1. Gli indici sintetici T^2 vengono dunque calcolati applicando, alle prime G componenti principali che si devono mantenere, la (3.3). Agli indici ottenuti viene infine applicato l'algoritmo delle k medie, con $K = 2$, e le $SSW(2)$ sono quindi ricavate, in accordo alla (3.4), sommando la devianza interna contenuta in ciascun gruppo.

Le 122 statistiche sono così calcolate su ogni video in controllo generato con il bootstrap a blocchi. Si fa notare che, la procedura per il calcolo delle statistiche, è implementata considerando subito tutti i $J = 161$ fotogrammi totali che compongono i video in oggetto. Questo metodo permette di calcolare inizialmente l'intera matrice di varianza-covarianza di dimensione $J \times J$ (161×161). Di questa matrice, in base all'indice $j = 1, \dots, J$, relativamente al quale si vuole calcolare la statistica, verrà considerata di volta in volta una sotto-matrice. Più chiaramente, la statistica $SSW_j(2)$, relativa al j -esimo fotogramma, viene ottenuta utilizzando, per il calcolo delle ST-PCA corrispondenti, la sotto-matrice di varianza-covarianza di dimensione $j \times j$. Questa sotto-matrice contiene gli elementi relativi ai primi j fotogrammi che si stanno considerando.

Si vuole sottolineare che questa procedura sarà utilizzata anche per lo studio di sorveglianza descritto nel prossimo capitolo, per calcolare le statistiche $SSW(2)$ sui video in cui è stata generata l'anomalia.

Seppure la metodologia descritta in questo elaborato sia stata proposta per la sorveglianza *in situ*, ovvero online, in cui non si ipotizza di avere subito a disposizione tutti i fotogrammi che compongono l'intero video, si è deciso di procedere in questo modo per giungere più velocemente ai risultati. Si tenga comunque presente che, per definire una funzione in grado di sorvegliare online le varie quantità, è sufficiente calcolare di volta in volta le

nuove covarianze, relative all'ultimo fotogramma registrato, come discusso nella Sezione 3.1, e confrontare quindi la statistica ottenuta con il limite precedentemente trovato.

Dalle statistiche $SSW_j(2)$ in controllo, vengono poi ottenute le relative E_j e W_j , applicando alle osservazioni $SSW_j(2)$ rispettivamente la (4.2) e la (4.3). Date le considerazioni fatte in Sezione 4.3, le statistiche EWMA e WCuSum vengono calcolate utilizzando per il parametro λ un valore pari a 0.6. Questa decisione è stata presa in quanto valori elevati del parametro λ sono preferibili quando si vogliono sorvegliare *shift* medio/grandi. Sono stati effettuati diversi studi di simulazione, che non vengono di seguito riportati, che hanno mostrato come queste carte, con valori di λ inferiori, non siano in grado di rilevare, come previsto, velocemente anomalie di medie e grandi dimensioni. Per quanto concerne il parametro κ della carta WCuSum, si è deciso di utilizzare $\kappa = 0.01$. Questo valore ridotto è stato scelto per due motivi: innanzitutto l'informazione relativa allo *shift* è in parte contenuta nel peso ϕ_j (Formula 4.4), che si aggiorna velocemente grazie al valore elevato di λ . In secondo luogo, si è voluto osservare come una carta disegnata per individuare piccoli cambiamenti, in processi in cui la media è dinamica, si comporti in contesti complessi e variabili come quello in oggetto.

Infine nella Appendice B.4 sono presentate le funzioni utilizzate per il calcolo dei limiti di controllo, attraverso il metodo di bisezione descritto in Sezione 4.2.2. I valori critici sono stati ottenuti per ogni tipo di statistica, per i seguenti valori nominali: $ARL_{IC} = \{30, 60, 90\}$. I limiti ottenuti sono riportati in Tabella A.1, e saranno utilizzati nel Capitolo 5 per sorvegliare i video su cui saranno generate 4 tipologie di anomalie.

Si vuole sottolineare che i valori di ARL_{IC} utilizzati sono piccoli, ma queste scelte sono state determinate dal ridotto numero di statistiche a disposizione. Per quanto discusso in Sezione 2.1, si ricorda che il valore di ARL_{IC} nominale, per cui si vogliono trovare i limiti di controllo, influisce, in generale, sulla reattività della carta quando applicata a campioni fuori controllo. Ci si aspetta, quindi, che i limiti calcolati sul valore nominale $ARL_{IC} = 30$ portino le carte a segnalare più allarmi e più velocemente.

Tabella 4.1: Valori delle \widehat{ARL}_{IC} (e relative deviazioni standard) osservate sui 500 video in controllo generati con il bootstrap a blocchi (limiti di controllo in Tabella A.1).

	$ARL_{IC} = 30$	$ARL_{IC} = 60$	$ARL_{IC} = 90$
SSW(2)	30.56 (44.66)	58.78 (55.82)	91.95 (49.97)
EWMA($\lambda = 0.6$)	32.06 (45.65)	57.87 (55.77)	88.86 (51.35)
WCuSum($\lambda = 0.6, \kappa = 0.01$)	39.68 (41.47)	56.41 (47.82)	88.84 (46.68)

4.5 Analisi dei risultati: performance in controllo delle carte

In questa sezione sono discusse le performance in controllo delle carte proposte. I valori \widehat{ARL}_{IC} sono calcolati, in accordo alla (2.2), come la media delle RL osservate per ciascun video in controllo. Queste sono ottenute, in accordo alla (2.1), come il primo istante $t = 1, \dots, 122$ in cui le statistiche eccedono i limiti di controllo riportati in Tabella A.1. I comandi utilizzati per il calcolo delle \widehat{ARL}_{IC} sono consultabili in Appendice B.5.

I risultati ottenuti, riportati in Tabella 4.1, confermano come i limiti calcolati con il metodo di bisezione garantiscano in media il numero di falsi allarmi desiderato a priori.

La Tabella 4.1 riporta anche le deviazioni standard. Innanzitutto si fa comunque notare che, per quanto concerne la carta SSW(2), che è una carta di tipo Shewhart, la RL ha una distribuzione geometrica con deviazione standard circa pari alla media (si veda Qiu, 2013). Quindi i risultati che si ottengono non si discostano eccessivamente dai risultati teorici, e la deviazione standard risulta anzi inferiore per un $ARL_{IC} = 90$ nominale. L'elevata variabilità osservata negli altri casi, può essere causata dai bassi valori di ARL_{IC} su cui è possibile calcolare i limiti di controllo, dato che si hanno a disposizione, per ogni video, solo 122 statistiche. Infine, come anticipato nella Sezione 4.2, l'utilizzo di un metodo di bisezione non tradizionale porta a risultati meno robusti, rispetto a quelli che si otterrebbero col metodo classico, tuttavia, per quanto già discusso, l'utilizzo del metodo tradizionale non sarebbe stato applicabile in questo contesto.

Dato che l'ARL_{IC} nominale viene in media garantito, per carte di controllo implementate in un contesto complesso, gli strumenti descritti saranno utilizzati per la sorveglianza di video rappresentativi del processo di fusione SLM. Nel prossimo e conclusivo capitolo saranno presentati i risultati relativi allo studio di simulazione effettuato, utilizzando i limiti di controllo riportati in Tabella A.1 e si mostreranno i diversi comportamenti delle carte, in presenza di diverse tipologie di anomalie.

Capitolo 5

Studio di simulazione

In questo capitolo conclusivo si condurrà un breve studio di simulazione per valutare le performance delle carte costruite, in termini di percentuale di anomalie rilevate e ARL_{OC} , durante la sorveglianza di video rappresentativi del processo fuori controllo. Questi saranno ottenuti simulando, sul video in controllo originale, anomalie termiche di diversa tipologia partendo da istanti differenti, utilizzando una versione modificata della funzione proposta da Colosimo e Grasso (2018). Si riporteranno infine le conclusioni ottenute e i possibili miglioramenti e le estensioni degli strumenti proposti.

5.1 Metodo di simulazione delle anomalie termiche

In questa sezione verrà descritta la funzione utilizzata per la simulazione di anomalie termiche.

Un *hot spot* è caratterizzato da un profilo di raffreddamento lento, il cui decadimento accelera col passare del tempo. Colosimo e Grasso (2018) mostrano come questa dinamica possa essere simulata modificando l'intensità di un numero ν di pixel adiacenti del video originale, per una determinata lunghezza temporale ω , attraverso l'utilizzo della funzione sigmoide di seguito riportata:

$$\tilde{u}_{m,n,j}(\omega) = 255 / (1 + e^{0.2((j-j_0+1)-0.95\omega)}), \quad j = j_0, \dots, (\omega + j_0) \quad (5.1)$$

Nella (5.1), j_0 indica il fotogramma in cui viene simulata l'insorgenza dell'anomalia, mentre j il fotogramma per cui si sta simulando l'intensità del pixel di coordinate (m, n) . I profili di raffreddamento simulati, $\tilde{u}_{m,n,j}$, attraverso la funzione riportata in Formula 5.1, riescono a riprodurre in modo soddisfacente i profili di anomalie reali. Tuttavia, la (5.1) non considera lo scenario in cui, durante il periodo di lunghezza ω , in cui si simula l'anomalia, il laser illumina nuovamente i ν pixel interessati dall'*hot spot*. Il laser può infatti passare più volte in una stessa area, e quando ciò avviene i pixel illuminati raggiungono la massima intensità a prescindere dallo stato che li caratterizzava all'istante precedente. Applicando la (5.1) questa dinamica non riesce ad essere rappresentata.

Supponiamo infatti di simulare una anomalia di durata $\omega = 30$, a partire dal fotogramma $j_0 = 45$, e che al 60-esimo fotogramma il laser illumini l'area in cui si trova l'*hot spot*. Il profilo dei pixel che si ottiene utilizzando la (5.1) è quella che si osserva nella Figura 5.1 (a sinistra). Tale profilo non è realistico della situazione presentata, in quanto al 60-esimo fotogramma l'intensità del pixel in oggetto dovrebbe essere massima. Per questo motivo è stata applicata una piccola modifica alla (5.1), in modo da superare questo ostacolo.

Il metodo di seguito proposto permette di simulare *hot spot* anche in scenari di questo tipo, definendo l'intensità dei pixel interessati dall'anomalia, condizionatamente all'intensità osservata nel video originale. In Figura 5.1 (a destra) è rappresentato il profilo simulato, col metodo di seguito proposto, nello scenario precedente, in cui al 60-esimo fotogramma il laser illumina la zona interessata dall'anomalia. Questa simulazione viene definita in due step successivi. Nella Formula 5.2 la funzione sigmoideale proposta da Colosimo e Grasso (2018) viene modificata, sottraendo alla (5.1) l'intensità media di *background*, pari a 80. Il valore $i_j(\omega)$ rappresenta così la luminosità aggiuntiva, che ci si aspetterebbe di osservare nel j -esimo fotogramma, quando una anomalia di durata ω insorge al fotogramma j_0 .

$$i_j(\omega) = 255 / (1 + e^{0.2((j-j_0+1)-0.95\omega)}) - 80, \quad j = j_0, \dots, (\omega + j_0) \quad (5.2)$$

In Formula 5.3, $\tilde{u}_{m,n,j}$ rappresenta invece l'intensità del pixel di coordinate (m, n) nel j -esimo fotogramma, in presenza di un *hot spot*, e viene ottenuta considerando il valore minimo tra 255, che è l'intensità massima osservabile,

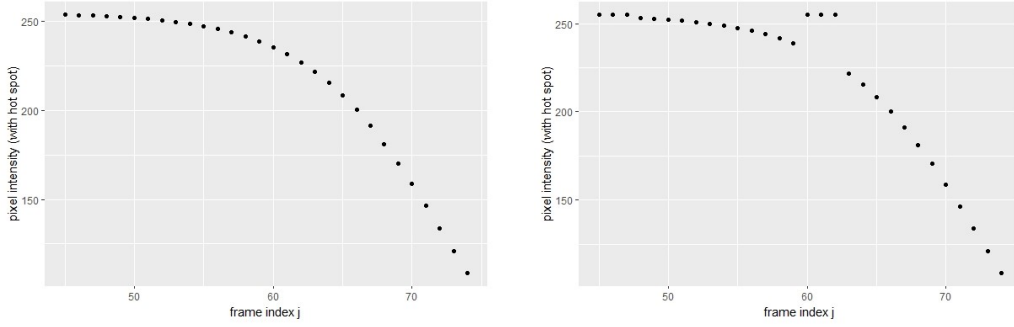


Figura 5.1: Profilo di raffreddamento di un *hot spot* di durata $\omega = 30$, insorto al 45-esimo fotogramma, simulato tramite la funzione sigmoide utilizzata da Colosimo e Grasso (2018) (a sinistra) e con la modifica proposta nelle Formule 5.2 e 5.3 (a destra). È stato ipotizzato il passaggio del laser sul pixel al 60-esimo *frame*.

e la somma dell'intensità originale $u_{m,n,j}$ con quella aggiuntiva simulata nella (5.2), $i_j(\omega)$.

$$\tilde{u}_{m,n,j}(\omega) = \min[255, u_{m,n,j} + i_j(\omega)], \quad j = j_0, \dots, (\omega + j_0) \quad (5.3)$$

Questo metodo di generazione di anomalie termiche, simulate direttamente su video in controllo, permette di tenere in considerazione tutti quegli scenari in cui il laser illumina le aree interessate dall'anomalia che si sta generando. Nel caso in cui non si sia in presenza di una situazione di questo tipo, allora in media, i valori di intensità del video originale sono circa pari a 80 (valore di *background*), per tutti gli ω fotogrammi successivi a j_0 e, di conseguenza, i profili simulati con la (5.1) e con le Formule 5.2 e 5.3 sono pressoché analoghi. In Figura 5.2 sono riportate 9 immagini, relative a 3 fotogrammi, rispettivamente di indici: $j = \{90, 100, 110\}$, estrapolati dal video originale, ovvero quello utilizzato per la simulazione del campione di 500 video in controllo. I fotogrammi riportati nella prima riga della figura sono quelli realmente osservati. I fotogrammi nelle successive due righe presentano invece un'anomalia termica, di breve durata ($\omega = 25$), e di media dimensione ($\nu = 45$ pixel), insorta all'85-esimo *frame*. In particolare, le immagini nella seconda riga sono state simulate con la funzione sigmoide utilizzata da Colosimo e Grasso (2018), mentre quelle riportate nell'ultima riga con il metodo proposto in questo elaborato. Si tenga presente che la forma "a croce" è stata utilizzata

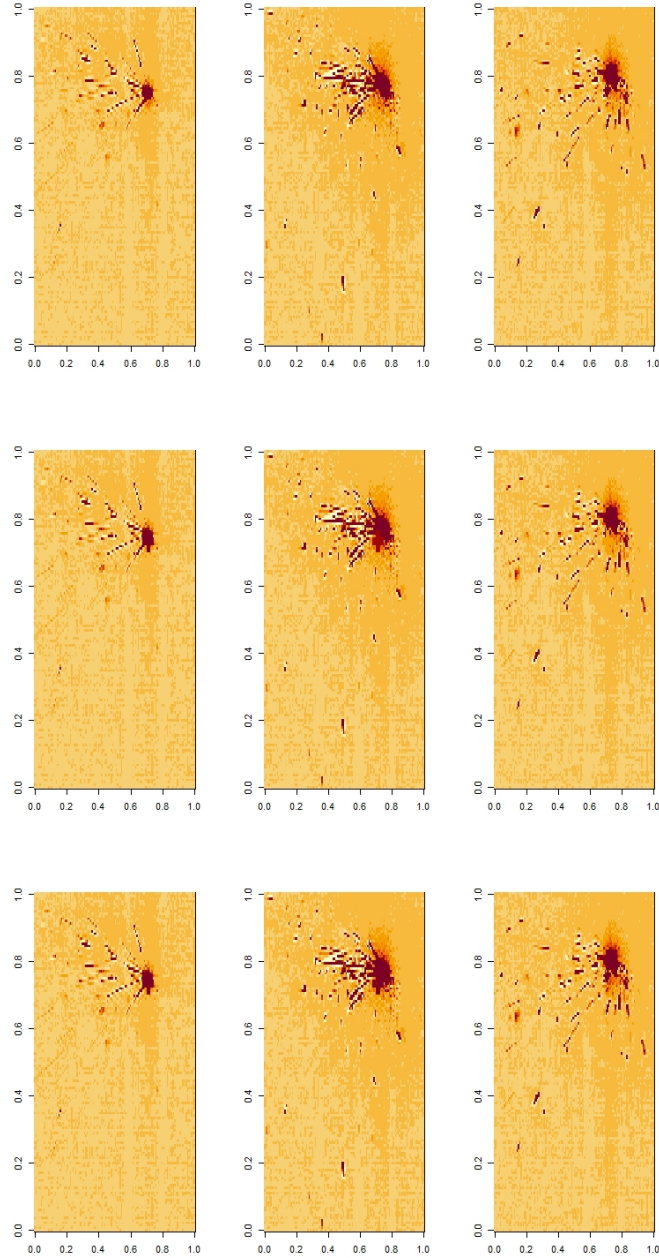


Figura 5.2: Fotogrammi di indici $j = \{90, 100, 110\}$, estrapolati dal video originale. La prima riga rappresenta le immagini realmente osservate, mentre la seconda e la terza riga rappresentano gli stessi fotogrammi in cui è stata però simulata una anomalia di media dimensione ($\nu = 45$ pixel), insorta all'85-esimo fotogramma, e di durata $\omega = 25$, utilizzando rispettivamente la Formula 5.1 (seconda riga), e le Formule 5.2 e 5.3 (terza riga).

a solo scopo illustrativo, col fine di evidenziare maggiormente l'*hot spot*, e che Colosimo e Grasso (2018) hanno dimostrato come la forma dell'anomalia non sia una caratteristica rilevante. È possibile osservare come, utilizzando la (5.1), i pixel interessati dall'anomalia, nel 100-esimo fotogramma (seconda immagine, della seconda riga), siano meno luminosi rispetto a quanto non si osservi nel fotogramma originale (seconda immagine, prima riga). Questo è causato dal fatto che il metodo di simulazione utilizzato non considera l'intensità dei pixel del video originale, e simula l'anomalia indipendentemente da ciò che si è osservato nella realtà. Questo problema non si riscontra invece nel fotogramma in cui l'anomalia è stata simulata con il metodo proposto (seconda immagine, terza riga). Qui i pixel che sono stati illuminati dal laser, durante la registrazione del video originale, conservano la massima intensità in seguito alla simulazione dell'*hot spot*.

5.1.1 Osservazioni sulla procedura di simulazione

Le funzioni implementate, impiegate per la simulazione di *hot spot*, sia utilizzando il metodo di Colosimo e Grasso (2018), sia quello proposto nelle (5.2) e (5.3), sono riportate in Appendice B.6.

È importante ricordare che gli *hot spot* sono aree di surriscaldamento, causate da un errato trasferimento di calore dal laser al materiale, e hanno dunque origine nelle zone di fusione. Per questo motivo, deciso il fotogramma j_0 dal quale si desidera dare inizio all'anomalia, i pixel corrispondenti sono individuati dalla posizione in cui si trovava il laser al momento della registrazione del fotogramma j_0 . Per questo motivo gli *hot spot* non possono essere simulati a partire da qualsiasi fotogramma, poiché non si hanno a disposizione tutte le posizioni del laser, ma solo quelle relative ai fotogrammi di indici $j = 11, \dots, 110$ del video originale (si veda `location`, Appendice A). Questo non permette quindi la simulazione delle anomalie a partire dai restanti fotogrammi, non essendo nota la posizione del laser in quegli istanti. Si fa comunque notare che l'insorgenza tardiva delle anomalie sarebbe anche difficilmente rilevabile, in quanto, dagli studi condotti da Colosimo e Grasso (2018), si è osservato come la rilevazione di anomalie di grandi dimensioni (con la configurazione di parametri riportata in Sezione 4.1, e l'utilizzo della

(5.1)) necessiti in media di circa 31 fotogrammi ($ARL_{OC} = 31$).

La mancata conoscenza della posizione del laser in determinati istanti del video originale rende inoltre difficoltosa la generazione delle anomalie sui video simulati tramite il bootstrap a blocchi. Per possibili studi futuri, nel caso in cui si volessero simulare anomalie termiche su qualsiasi video generato a partire da un video originale, è stata implementata una funzione (Appendice B.6) che permette di conoscere la posizione del laser nei fotogrammi di ciascun video, combinando le posizioni del laser rilevate sul video originale con gli indici dei fotogrammi che compongono i video simulati.

In questo modo è possibile simulare anomalie di dimensioni, e durate differenti su qualsiasi video a disposizione.

Come già anticipato la dimensione dell'anomalia è determinata dal numero ν di pixel che la costituiscono, e il pixel centrale viene appunto individuato dalle coordinate del laser al j_0 -esimo fotogramma. In base alla forma e alla dimensione che si vogliono dare all'anomalia, i restanti pixel sono individuati come quelli adiacenti al pixel centrale. Nella prossima sezione sarà condotto uno studio di simulazione sul video originale, generando 4 differenti tipologie di anomalie termiche.

5.2 Studio di simulazione

In questa sezione verrà descritto lo studio di simulazione effettuato generando anomalie sul video originale.

Si è deciso di simulare 4 tipologie di anomalie, determinate dalla combinazione di 2 categorie di dimensione e 2 valori per la durata ω . In particolare, verranno simulate anomalie di dimensione $\nu = \{45, 80\}$ (rispettivamente: medie/grandi e grandi), e di durata $\omega = \{30, 50\}$ (rispettivamente di media e lunga durata). Per le altre tipologie di *hot spot*, che possono essere simulate, si rimanda all'Appendice B.6.

In particolare, per ognuna delle 4 tipologie di *hot spot*, sono stati simulati 30 video fuori controllo, dove l'anomalia viene fatta insorgere a partire da valori j_0 , campionati casualmente tra i 100 disponibili. Si ricorda infatti che si possono generare le anomalie solo a partire dai fotogrammi di indice:

$j = 11, \dots, 110$.

Si ottengono quindi i seguenti scenari fuori controllo:

- A_1 : scenario in cui sono state simulate anomalie di dimensione $\nu = 80$ e durata $\omega = 50$ (grandi e di lunga durata);
- A_2 : scenario in cui sono state simulate anomalie di dimensione $\nu = 80$ e durata $\omega = 30$ (grandi e di media durata);
- A_3 : scenario in cui sono state simulate anomalie di dimensione $\nu = 45$ e durata $\omega = 50$ (medie e di lunga durata);
- A_4 : scenario in cui sono state simulate anomalie di dimensione $\nu = 45$ e durata $\omega = 30$ (medie e di media durata);

In Appendice B.6 è consultabile il codice utilizzato per la generazione dei video fuori controllo, e per il calcolo delle relative statistiche SSW(2), EWMA e WCuSum.

Le statistiche calcolate sono quindi confrontate con i limiti di controllo riportati in Tabella A.1, e sono così ottenute le RL delle carte (Formula 2.1). I risultati delle performance delle carte sono riassunti nella Tabella 5.2, in cui sono riportati: l'ARL_{OC}, la percentuale di anomalie rilevate e la relativa deviazione standard, per ogni scenario: A_1, A_2, A_3, A_4 di video fuori controllo. Si tenga presente che l'ARL_{OC} è calcolato aggiungendo alla (2.3) i 40 fotogrammi di *batch*, così da ottenere il numero di fotogrammi necessari per chiamare l'allarme.

Il codice per il calcolo di tali quantità è consultabile in Appendice B.7.

Si fa presente che nessuna carta, applicata al video originale, ha segnalato allarmi, coerentemente con lo stato in controllo del processo.

Innanzitutto, dai risultati ottenuti è possibile affermare che le carte di controllo reagiscono in modo diverso ai tipi di anomalia simulati, e che, all'aumentare dell'ARL_{IC} nominale, la reattività delle carte (ARL_{OC}), e soprattutto l'efficienza nell'individuare l'anomalia (in percentuale di anomalie individuate), diminuiscono.

Come ci si poteva aspettare, date le considerazioni fatte nella Sezione 4.3, la carta SSW(2) di tipo Shewhart, è quella che risulta in generale più sensibile

alle anomalie simulate, in particolare a quelle di grandi dimensioni e di lunga durata: riporta i valori più bassi di ARL_{OC} e il maggior numero di *hot spot* rilevati, con i limiti calcolati su valori nominali $ARL_{IC} = \{30, 60\}$.

Si può inoltre osservare come la carta SSW(2) e la carta EWMA conducano a risultati molto simili. Questo è motivato dal valore elevato utilizzato per il parametro λ , che porta la carta con memoria ed essere maggiormente reattiva ai grandi cambiamenti.

Particolarmente interessanti sono i risultati ottenuti per lo scenario A_2 , su cui sono state simulate anomalie di grandi dimensioni e di media durata. In questo caso la percentuale di anomalie rilevate dalle carte diminuisce visibilmente, rispetto a quella ottenuta nello scenario A_1 , fino quasi ad annullarsi con i limiti calcolati garantendo un ARL_{IC} nominale pari a 90. La WCuSum, in particolare, non riesce quasi mai a segnalare la situazione di fuori controllo in questo caso. Confrontando poi i risultati ottenuti per lo scenario A_3 , contro quelli ottenuti per lo scenario A_2 , è possibile notare come la durata ω abbia un effetto più rilevante sulle statistiche, rispetto alla dimensione dell'anomalia. Questa osservazione trova spiegazione nell'utilizzo delle ST-PCA come metodo di riduzione della dimensionalità: maggiore è la durata dell'anomalia, maggiore sarà il segnale, relativo al profilo di raffreddamento anomalo, contenuto nelle statistiche. Inoltre, il fatto che le anomalie di media dimensione e di lunga durata siano mediamente quelle più velocemente rilevabili, da tutte e tre le carte, e che siano quelle la cui percentuale di anomalie individuate rimane circa stabile, è un risultato interessante, e che potrebbe essere oggetto di studi futuri.

Particolarmente rilevanti sono i risultati ottenuti con la WCuSum. Si può infatti osservare come questa carta sia, tra tutte, quella più robusta in termini di differenza di percentuale di anomalie rilevate e ARL_{OC} per i diversi valori di ARL_{IC} nominali. Questo la rende uno strumento valido anche quando la dimensione del cambiamento non è piccola. Inoltre si può osservare come, nello scenario A_4 , relativo alle anomalie di media grandezza e durata, la WCuSum sia la carta che riesce a segnalare più anomalie (con i valori di $ARL_{IC} = \{60, 90\}$). Studi più approfonditi possono essere condotti per valutare le performance di questa carta in presenza di anomalie di dimensioni e durate inferiori. Si crede infatti che, studiando più approfonditamente la

Tabella 5.1: Risultati ottenuti dallo studio di simulazione: Si riportano, per ogni campione simulato: A_1 , A_2 , A_3 e A_4 i valori di ARL_{OC} , calcolati in accordo alla (2.3), e le relative percentuali di anomalie rilevate e deviazione standard, per le 3 carte di controllo costruite (limiti in Tabella A.1).

		$ARL_{IC} = 30$	$ARL_{IC} = 60$	$ARL_{IC} = 90$
A_1	SSW(2)	27.96(100%, 19.84)	39.41(80%, 21.24)	85.72(36%, 38.58)
	EWMA	29.73(100%, 21.54)	40.04(80%, 20.78)	77.92(46%, 32.97)
	WCuSum	41.06(63%, 22.11)	41.25(53%, 21.84)	60.26(50%, 23.75)
A_2	SSW(2)	20.09(73%, 12.99)	28.56(53%, 14.04)	67(6%, 49.97)
	EWMA	20.72(73%, 12.99)	29.25(53%, 14.07)	80(3%)
	WCuSum	80(3%)	80(3%)	80(3%)
A_3	SSW(2)	22.17(96%, 8.04)	26.09(70%, 7.61)	48(60%, 36.76)
	EWMA	22.68(96%, 7.98)	27.18(73%, 7.89)	43(60%, 23.59)
	WCuSum	38.89(63%, 29.55)	39.52(56%, 30.71)	45(53%, 27.83)
A_4	SSW(2)	29.80(70%, 17.14)	25.45(36%, 7.63)	29(10%, 7.21)
	EWMA	30.42(70%, 17.02)	25.63(36%, 7.61)	25.66(10%, 2.88)
	WCuSum	40.20(66%, 24.61)	31.64(46%, 20.26)	25.62(26%, 9.75)

regolazione dei parametri, e applicando la carta WCuSum in contesti in cui vengono simulati piccoli *shift*, si possano ottenere risultati soddisfacenti.

Si può infine osservare come la variabilità delle RL sia elevata. Questo fenomeno è attribuibile all'influenza che hanno gli istanti j_0 , relativi all'insorgenza dell'anomalia, sull'aumento delle $SSW(2)$, ovvero l'insorgenza di una stessa anomalia, a partire da fotogrammi differenti, non conduce allo stesso aumento nelle osservazioni $SSW_j(2)$, questo risultato è ovviamente influenzato dal fatto che la metodologia ST-PCA, per poter essere implementata, necessita di un *batch* di 40 fotogrammi iniziali.

In generale, i risultati ottenuti confermano quelli riportati da Colosimo e Grasso (2018) per *hot spot* di grandi dimensioni (relativi agli *hot spot* di media dimensione qui simulati). Ma, in aggiunta, la metodologia proposta in questo elaborato permette di determinare dei limiti di controllo, con un vincolo sul numero medio di falsi allarmi. È inoltre probabile che il differente metodo di simulazione di *hot spot* influenzi le statistiche con le quali si sorveglia il processo. Per questo motivo, studi futuri possono valutare le differenti performance dei sistemi di sorveglianza, simulando video fuori controllo con le due metodologie proposte.

5.3 Sviluppi futuri

La procedura proposta in questo elaborato ha condotto ad alcuni risultati interessanti, lasciando anche diverse domande aperte a possibili sviluppi futuri, per l'implementazione di sistemi di sorveglianza strutturati, nell'ambito dei processi SLM, che siano sempre più efficienti. Di seguito sono proposte alcune possibili estensioni.

In particolare, si è potuto mostrare empiricamente, con lo studio di simulazione proposto in Sezione 5.2, come carte differenti non siano sensibili allo stesso modo a diverse tipologie di cambiamento. Si potrebbero, per questo motivo, implementare carte di controllo combinate, in grado di rilevare il prima possibile tipologie di anomalie differenti. In questo elaborato, si è voluto dare particolare importanza alla rilevazione di anomalie di medie/grandi dimensioni, ma anche in questo caso si è osservato come variazioni nella durata e nella grandezza dell'*hot spot* abbiano condotto a risultati molto diversi.

È stato inoltre mostrato come l'utilizzo di metodologie che tengano conto della dipendenza temporale di fotogrammi successivi, come il metodo di riduzione delle ST-PCA, sia indispensabile in contesti di questo tipo.

Si fa notare che la metodologia proposta da Colosimo e Grasso (2018), e implementata in questo elaborato, permette però il fenomeno di *label switching* nei pixel, ovvero pixel che al j -esimo fotogramma vengono raggruppati in un gruppo, al $(j + 1)$ -esimo fotogramma possono essere contenuti in un altro. Questo fenomeno è essenziale affinché le anomalie possano essere rilevate, in quanto la dinamica dei pixel si evolve con il passare del tempo, ed è quindi necessario che le unità abbiano la possibilità di passare da un *cluster* all'altro durante la sorveglianza. Potrebbe però essere interessante sviluppare una metodologia che permetta, ad ogni istante, di verificare in che gruppo sono contenuti i singoli pixel, tracciandone la loro corsa. Questo permetterebbe di avere una visione più dettagliata della dinamica del processo di fusione e consentirebbe di sviluppare carte di controllo entro i gruppi.

Conclusioni

Il continuo progresso dei processi di produzione additiva ha permesso di ottenere prodotti sempre più complessi, e la crescente diffusione di questa tecnologia ha reso necessario lo sviluppo di metodologie di sorveglianza avanzate in grado di controllare la stabilità dei processi. Questi ultimi, infatti, sono interessati da differenti tipologie di anomalie, che possono insorgere durante le diverse fasi della produzione.

Il presente elaborato si è posto l'obiettivo di sviluppare una metodologia di SPC per la sorveglianza di dati video, rappresentativi della fase di fusione di uno strato, nei processi SLM. In particolare, è stata definita una procedura che, attraverso l'utilizzo di un bootstrap a blocchi circolare, e l'applicazione di un metodo di bisezione non tradizionale, ha permesso di calcolare dei limiti di controllo che consentono di segnalare se, durante la procedura di fusione dello strato, è presente un'anomalia termica, con un vincolo sul numero medio di falsi allarmi.

Il metodo proposto sfrutta le ST-PCA introdotte da Colosimo e Grasso (2018), le quali si è mostrato essere di estrema efficacia per catturare la dinamica che caratterizza i dati video, e per ridurre il rumore spesso presente in questi dati complessi. In questo caso le ST-PCA sono state utilizzate per calcolare le statistiche $SSW(2)$, che riassumono l'eterogeneità nei profili dei pixel.

Si è quindi discusso di come il processo di fusione SLM possa essere sorvegliato attraverso l'utilizzo di tre statistiche: $SSW(2)$, EWMA e $WCuSum$, per ognuna delle quali sono stati calcolati i limiti di controllo con il metodo di bisezione. Si è dimostrato come questi riescano a garantire, per ogni statistica, un certo numero medio di falsi allarmi, per diversi valori dell' ARL_{IC}

nominale, e si sono inoltre discussi i motivi principali che rendono debolmente robusta questa metodologia.

È stato quindi mostrato, con uno studio di simulazione, come le carte proposte riescano, in generale, a rilevare le anomalie, simulate attraverso una versione modificata della funzione sigmoideale proposta da Colosimo e Grasso (2018), e si è potuto, infine, vedere come la reattività delle carte sia fortemente influenzata dal tipo di statistica utilizzata per sorvegliare il processo. Il presente studio ha quindi contribuito all'ambito della sorveglianza statistica dei processi di produzione additiva, sviluppando un metodo che permette di ottenere rapidamente dei limiti, che sono in grado di indicare se il processo esaminato è stabile o meno, per qualsiasi statistica si voglia utilizzare. Ulteriori studi potrebbero seguire questa direzione al fine di superare alcuni limiti del presente studio, proponendo, ad esempio, statistiche differenti in grado di descrivere in modo più appropriato il processo.

Appendice A

Elementi per l'implementazione della metodologia

In questa Appendice sono elencati gli elementi utilizzati nelle funzioni (riportate in Appendice B), per l'implementazione della metodologia proposta. In Tabella A.1 sono riportati i limiti di controllo ottenuti.

- **dataIC**: video originale, a cui sono stati rimossi il primo e gli ultimi 13 fotogrammi, in formato *array*, di dimensione $126 \times 136 \times 161$ (gentilmente concesso da Marco Luigi Giuseppe Grasso).
- **Index.video**: Lista di 500 vettori. L' i -esimo vettore, con $i = 1, \dots, 500$ contiene gli indici j dei fotogrammi, relativi al video originale **dataIC**, che compongono l' i -esimo video (ottenuta con i comandi consultabili in Appendice B.2).
- **Index.HS**: Lista di 4 vettori. Il q -esimo vettore, con $q = \{1, 2, 3, 4\}$ contiene gli istanti j_0 di insorgenza dell'anomalia nel q -esimo campione di video simulati. In particolare il primo campione è relativo allo scenario A_1 , il secondo allo scenario A_2 e così via (ottenuta con i comandi consultabili in Appendice B.6).
- **location**: matrice di dimensione 2×100 . Contiene le coordinate del centroide del laser nei fotogrammi $j = 11, \dots, 110$ (proveniente dal file `location.txt`, gentilmente concesso da Marco Luigi Giuseppe Grasso).

- `sims.ewma`: matrice 500×122 . L' i -esima riga della matrice contiene le statistiche EWMA calcolate sull' i -esimo video generato con il metodo del bootstrap a blocchi (ottenuta con i comandi consultabili in Appendice B.4).
- `sims.ssw`: matrice 500×122 . L' i -esima riga della matrice contiene le statistiche SSW(2) calcolate sull' i -esimo video generato con il metodo del bootstrap a blocchi (ottenuta con i comandi consultabili in Appendice B.4).
- `sims.wcus`: matrice 500×122 . L' i -esima riga della matrice contiene le statistiche WCuSum calcolate sull' i -esimo video generato con il metodo del bootstrap a blocchi (ottenuta con i comandi consultabili in Appendice B.4).
- `SSW2.0C`: lista di 4 elementi. Ogni elemento consiste in una lista relativa ai 4 scenari simulati (descritti in Sezione 5.2). Ognuna di queste contiene 30 vettori di 122 statistiche SSW(2) calcolate sui relativi video fuori controllo, generati con la funzione `hotspot.fun` (Appendice B.6) (ottenuta con i comandi riportati in Appendice B.6).
- `video.IC`: generico video $i = 1, \dots, 500$ simulato con il bootstrap a blocchi, di dimensione $126 \times 136 \times 161$ (ottenuti con i comandi riportati in Appendice B.2).
- `W.mat`: matrice dei pesi, di dimensione $p \times p = 17136 \times 17136$, relativa alla matrice \mathbf{W}_1 (3.2) (calcolata utilizzando le funzioni in Appendice B.3).

Tabella A.1: Limiti controllo relativi alle carte SSW(2), EWMA e WCuSum, calcolati per tre valori di ARL_{IC} nominale.

	SSW(2)	EWMA($\lambda = 0.6$)	WCuSum($\lambda = 0.6, \kappa = 0.01$)
$ARL_{IC} = 30$	0.3248	0.3164	0.00024
$ARL_{IC} = 60$	0.4193	0.4173	0.00237
$ARL_{IC} = 90$	0.5517	0.5497	0.010

La Tabella A.1 contiene i limiti di controllo calcolati per le tre statistiche: SSW(2), EWMA e WCuSum per i tre valori di $ARL_{IC} = \{30, 60, 90\}$ nominali. Il `seed` è stato posto pari a 0000, e i risultati sono stati ottenuti utilizzando la funzione `bisez.fun` e i comandi riportati in Appendice B.4, con i seguenti valori per i parametri:

- Numero massimo di iterazioni: 20.
- Accuratezza ρ : 0.0001.

Appendice B

Codice R implementato

Nelle sezioni di questa Appendice sono descritti, e riportati, i comandi e le funzioni R utilizzati per l'implementazione della metodologia proposta in questo elaborato, e per l'acquisizione degli oggetti riportati in Appendice A.

B.1 Manipolazione delle strutture video

Di seguito vengono riportate le funzioni per l'*unfolding* e il *folding* dei dati video (Figura 3.2), rispettivamente `unfolding.fun` e `folding.fun`.

Gli unici oggetti necessari in input sono: (i) i dati video: in struttura *folded* per la funzione `unfolding.fun`, e in struttura *unfolded* per la funzione `folding.fun`, e (ii) le dimensioni dell'*array* che si desidera in output per la funzione `folding.fun`. Queste funzioni saranno più volte richiamate e utilizzate nelle varie fasi dell'implementazione della metodologia proposta.

```
1 # Funzione per l'unfolding di un generico video
2 # in INPUT:
3 # - dati: video in formato array (folded)
4
5 unfolding.fun = function(dati)
6 {
7   dim.dati = dim(dati)
```

```
8  video.unfolded = matrix(0, nrow = dim.dati[1]*dim.dati[2], ncol =
    dim.dati[3])
9  for(i in 1:dim.dati[3]){
10     video.unfolded[, i] = as.vector(dati[, ,i])
11 }
12 return(video.unfolded)
13 }
14
15
16 # Funzione per il folding di un generico video
17 # in INPUT:
18 # - dati.unf: video in formato matriciale (unfolded)
19 # - DIMx: dimensione x del nuovo array che si vuole in output
20 # - DIMy: dimensione y del nuovo array che si vuole in output
21
22 folding.fun = function(dati.unf, DIMx, DIMy)
23 {
24     IM = array(NA, dim = c(DIMx, DIMy, ncol(dati.unf)))
25     for (t in 1:ncol(dati.unf))
26     {
27         k <- 1
28         for (jj in 1:DIMy)
29         {
30             for (ii in 1:DIMx)
31             {
32                 IM[ii,jj,t] <- dati.unf[k, t]
33                 k <- k + 1
34             }
35         }
36     }
37     return(IM)
38 }
```

B.2 Implementazione del bootstrap a blocchi

Il codice di seguito riportato è stato utilizzato per ottenere il campione di 500 video in controllo (si veda l'oggetto `video.IC`, Appendice A), con l'implementazione del bootstrap a blocchi circolare, discusso in Sezione 4.2.1.

In particolare, i valori $b_{opt,d}$ vengono calcolati utilizzando la funzione `b.star`.

Questa funzione è disponibile al seguente indirizzo: <https://github.com/JeffreyRacine/R-Package-np/blob/master/R/b.star.R>.

Si tratta del codice originale in Matlab implementato da A. Patton, R (Patton, Politis e White, 2009) e tradotto in R, basato sull'articolo originale di Politis e White (2004).

Non sono state apportate modifiche al codice. La funzione `b.star` prende in input i dati, e i parametri legati al tipo di stima che si vuole ottenere. Nel nostro caso consideriamo tutti i valori di default. Si tenga presente che i dati devono essere in formato *unfolded* e trasposti, in quanto la funzione, per ogni variabile rilevata, calcola il valore ottimale $b_{opt,d}$. La funzione `b.star` riporta in output due lunghezze ottimali. Noi utilizziamo la seconda, che è relativa al metodo CB di nostro interesse.

Infine, vengono riportati i comandi per ottenere la Figura 4.1 e per la generazione dei video in controllo, attraverso la procedura bootstrap, con blocchi di lunghezza $b_{opt} = 20$. Con i comandi riportati vengono campionati, per ogni blocco, l'indice del fotogramma iniziale che darà inizio alla sequenza dei 20 fotogrammi, ipotizzando una struttura circolare degli indici. Campionati gli indici che definiranno il nuovo video, questo viene ottenuto selezionando i relativi fotogrammi dal video originale (`dataIC`, Appendice A). Gli indici che compongono ciascun video sono salvati e riportati nella lista `Index.video` (Appendice A).

```
1 # Funzione per il calcolo della lunghezza ottimale del
2 # blocco per ciascun profilo.
3
4 # Utilizzo della funzione b.star
5
6 # Di seguito vengono riportati i comandi utilizzati
7 # per la generazione dei 500 video utilizzando una
8 # lunghezza di blocco b_opt = 20.
```

```
9 # Si e' osservato (Sez 4.2) che questa lunghezza garantisce
10 # la copertura della correlazione nel 98% dei profili
11
12 # bstar ritorna tutti i b ottimali che riescono a
13 # mantenere la correlazione del singolo pixel
14 # tdataic.unfolded: matrice unfolded del video
15 # originale trasposta
16 bstar = b.star(data = tdataic.unfolded)
17
18 # Comandi per ottenere il grafico in Figura 4.1
19 plot(bstar[,2], type = "l", ylab = "Optimal Bootstrap block length",
20      xlab = "Pixel index")
21 abline(h = 20, col = 2, lty = 3)
22 mean(bstar[,2] <= 20)
23
24
25
26 # Comandi per la generazione dei video (videoIC)
27 # con il bootstrap a blocchi di lunghezza b = 20,
28 # l'oggetto dataic e' l'array che descrive il
29 # video originale, di dimensione 126 x 136 x 161
30 # Tutti i video generati avranno la
31 # stessa dimensione del video originale
32
33
34 block_length = 20
35 # lista che conterra' i vettori degli indici dei
36 # frames che compongono ciascun video
37 Index.video = list()
38 set.seed(123)
39
40 # creiamo 500 video IC
41 for(i in 1:500){
42   cat("\n", v)
43   idx = rep(0,161)
44   for(b in 1:9){ # Ogni video e' composto da 9 blocchi
```

```
sequenziali
45 start = sample(1:161, 1) # Campioniamo il primo frame di ciascun
    blocco
46
47 # Se abbiamo raggiunto l'ultimo blocco,
48 # questo e' formato dal solo indice di start
49 if(b == 9) {idx[161] = start}
50 # Se non abbiamo raggiunto l'ultimo blocco
51 # prendiamo tutti gli indici successivi
52 # a quello di start
53 else{
54
55 # In questo modo forziamo una struttura circolare,
56 # coscche' se ho raggiunto
57 # il frame di indice 161, riparto dal primo
58 idx[(1 + (b-1)*block_length):(b*block_length)] =
59     (1 + start:(start + block_length - 1) %% 161)
60 }
61 }
62
63 Index.video[[i]] = idx # Ogni elemento della lista
64                       # contiene gli indici idx
65
66 videoic = dataic[, ,idx] # Generiamo l'i-esimo video
67 save(videoic, file = paste0("videoIC", i, ".RData"))
68 }
```

B.3 Calcolo delle Statistiche di controllo

In questa sezione sono riportate le funzioni per il calcolo delle statistiche SSW(2), WCUsum e EWMA. ¹ Il codice per il calcolo della matrice di pesi

¹Le funzioni: `coord.fun`, `dist_xy`, `dist.mat.fun`, `w1.fun` e `ssw2.fun` sono le versioni modificate di quelle presenti nella tesi di Laurea Magistrale in Scienze Statistiche "Sorveglianza statistica di immagini: confronto tra due approcci di riduzione dimensionale" (Tasinato, anno 2022/2023, UniPd).

`w.mat` (\mathbf{W}_1 , Formula 3.2), è organizzato in 4 funzioni che permettono il calcolo dei seguenti elementi: (i) matrice $p \times 2$ delle coordinate (m, n) dei pixel, (ii) distanza euclidea tra due pixel di coordinate date, (iii) matrice contenente le distanze euclidee tra tutti i pixel, e (iv) matrice dei pesi `w.mat`.

La matrice `w.mat` è poi presa in input dalla funzione `ssw2.fun` che calcola le statistiche di tipo Shewhart SSW(2). La funzione `ssw2.fun` calcola la matrice di varianza-covarianza \mathbf{S} dell'intero video dato in input (*unfolded*), e, ad ogni iterazione (ad ogni nuovo *frame* che viene considerato) calcola le ST-PCA, utilizzando la sotto-matrice relativa ai fotogrammi corrispondenti, in accordo con quanto descritto nella Sezione 4.4. Di conseguenza si ottengono quindi, ad ogni iterazione, le statistiche T^2 , e la relativa SSW(2). La funzione è stata implementata sia per un aggiornamento ricorsivo dei dati, sia per un aggiornamento a finestra mobile, con la possibilità di regolare la finestra L e la percentuale di varianza spiegata delle componenti principali.

Dalle statistiche SSW(2) ottenute vengono quindi calcolate le relative EWMA e WCuSum.

La funzione `ewma.fun` calcola le statistiche EWMA (4.2), prendendo in input le statistiche SSW(2) date in output dalla funzione `ssw2.fun`, e il valore del parametro λ .

Infine le funzioni `wcus.fun` e `pesi.fun` sono utilizzate per il calcolo delle statistiche WCuSum. In particolare, la funzione `wcus.fun` richiama `pesi.fun`, la quale, dati in input le statistiche SSW(2) e il valore di λ , calcola i pesi della WCuSum (4.4).

```

1 # Funzione per il salvataggio delle coordinate
2 # dei pixel in un'immagine
3 # in INPUT:
4 # - immagine: matrice relative ad
5 #           una qualsiasi immagine
6 coord.fun = function(immagine){
7   cbind(riga = rep(1:nrow(immagine), ncol(immagine)),
8         colonna = sort(rep(1:ncol(immagine), nrow(immagine))))
9 }
10
11
```

```
12 # Funzione per il calcolo della distanza euclidea
13 # tra due pixel di coordinate (x_1, x_2) e (y_1, y_2)
14 # in INPUT:
15 # - x: coordinate del primo pixel
16 # - y: coordinate del secondo pixel
17 dist_xy = function(x,y){
18   as.numeric(((x[1] - y[1])^2 + (x[2]-y[2])^2)^0.5)
19 }
20
21
22 # Funzione per ottenere W0.mat: W_0:
23 # matrice delle distanze euclidee tra i pixel
24 # in INPUT:
25 # - coord: matrice di coordinate dei punti
26 #         da inserire nel grafico:
27 #         - prima colonna = coordinate di RIGA
28 #         - seconda colonna = coordinate di COLONNA
29
30 dist.mat.fun = function(coord){
31   w = matrix(0, ncol = nrow(coord), nrow = nrow(coord))
32   cat(paste0("\nInizio del calcolo di: ", sum(1:nrow(coord)-1)),
33       " distanze \n ",
34       " Il ciclo termina dopo: " ,(nrow(coord) -1)," iterazioni \n")
35   for(i in 1: (nrow(coord)-1)){
36     for(j in (i+1): nrow(coord)){
37       w[i,j] = w[j,i] =
38         ((coord[i,1]- coord[j,1])^2+(coord[i,2] - coord[j,2])^2)^0.5
39     }
40   }
41   return(w)
42 }
43
44
45 # Funzione per il calcolo della matrice
46 # di distanza W.mat: W_1
47 # in INPUT:
```

```
48 # - dist.mat: matrice delle distanze W0.mat
49 #           ritornata da dist.mat.fun
50
51 w1.fun = function(dist.mat, scaled = T){
52   cat(paste0("Calcolo di: W_1"))
53   # W_1
54   temp = (1/(dist.mat^2))
55   diag(temp) = 0
56   if(scaled) {cat("\nScaling..."); temp = temp/sum(temp)}
57   return(temp)
58 }
59
60
61 # Otteniamo la matrice W.mat non scalata
62 coord = coord.fun(dataic)
63 W0.mat = dist.mat.fun(coord)
64 W.mat = w.fun(W0.mat, scaled = F)
```

```
1 ## STATISTICHE SSW(2), EWMA e WCUSUM
2
3 # Dato il costo computazionale,
4 # le statistiche
5 # vengono definite per una sola combinazione
6 # di parametri:
7 # 1) metodo ricorsivo
8 # 2) 50% di varianza spiegata per le PCA
9 # 3) matrice di pesi W = W1
10
11
12 # Funzione per il calcolo delle SSW(2) con il
13 # metodo ST-PCA proposto da Colosimo e Grasso
14 # La seguente funzione non e' stata implementata
15 # per la sorveglianza di video online
16 # ma solo per il calcolo delle statistiche SSW2
17 # (sia sui video IC che OC)
18 # in INPUT:
```



```
19 # - data.unf: matrice unfolded del video su
20 #           cui calcolare le statistiche
21 # - W.mat: matrice delle distanze continua
22 # - soglia_pc: soglia di varianza spiegata
23 #           per cui calcolare le componenti
24 #           principali (0.5 di default)
25 # - MW: metodo a finestra mobile (FALSE di default)
26 # - L: parametro per la lunghezza della finestra mobile
27 # - n_frame_start: lunghezza del batch di
28 #           fotogrammi iniziale (40 di default)
29
30 ssw2.fun = function(data.unf = NULL, MW = F, L = 50, soglia_pc = 0.5,
31                   W.mat = W.mat,
32                   n_frame_start = 40, seed = 123)
33 {
34   set.seed(seed)
35   # controllo iniziale
36   if(is.null(data.unf)){
37     warning(paste("SSW2: controllare data.unf: presente valore NULL"))
38     return(NULL)
39   }
40   if(MW) cat("\nMetodo Moving Window con finestra mobile L: ",L)
41   else cat("\nMetodo Ricorsivo")
42
43   # dimensioni dei video
44   nframes = dim(data.unf)[2]
45   p = dim(data.unf)[1]
46
47   # Partiamo dal frame n_frame_start
48   frame.index = n_frame_start
49   # Inizializzazione del vettore
50   # che conterra' le statistiche ssw2
51   SSW2 = rep(NA, nframes - n_frame_start + 1)
52   # Riscaliamo i pixel in (0,1)
53   cat("\nRiscaliamo i pixel tra (0,1)")
54   dati = (data.unf/255)
```

```

54
55 # Calcolo della matrice S completa
56 # di dimensione J x J (nframes x nframes)
57
58 S = (1/(p-1)) * t(apply(dati, 2, function(x) {x- mean(x)})) %*%
59   W.mat %*% apply(dati, 2, function(x) {x- mean(x)})
60 cat("\n\nInizio ciclo While per il calcolo delle ",(nframes -
    n_frame_start+1), "statistiche")
61
62 # Inizio del ciclo per calcolare le statistiche
63 # ad ogni nuovo fotogramma. Il ciclo
64 # si interrompe al raggiungimento frame finale
65 # Ad ogni iterazione prendiamo solo la "porzione"
66 # di S relativa ai frames che ci interessano e
67 # ci applichiamo la decomposizione spettrale
68
69 while(frame.index <= nframes){
70
71   # Se il metodo e' MW e frame.index > L
72   if(frame.index > L && MW == T){
73     SD = eigen(S[(frame.index - L):frame.index , (frame.index - L)
74       :frame.index])
75   }
76   # Se frame.index e' <= L o il metodo e' ricorsivo
77   else{
78     SD = eigen(S[1:frame.index,1:frame.index])
79   }
80
81   lambda = SD$values # autovalori
82   eig = SD$vectors   # autovettori
83
84   # Percentuale di varianza spiegata dalle CP
85   perc.var.pc = lambda/sum(lambda)
86
87   # Numero di cp da conservare
88   n.pc = min(which(cumsum(perc.var.pc) >= soglia_pc))

```

```
88
89 # Calcoliamo le componenti principali Z
90 # Se frame.index > L e il metodo e' MW
91 if(frame.index > L && MW == T){
92     Z = dati[, (frame.index - L):frame.index] %*% eig[,1:n.pc]
93 }
94 # Se frame.index <= L o il metodo e' Ricorsivo
95 else{
96     Z = dati[,1:frame.index] %*% eig[,1:n.pc]
97 }
98
99 # Calcoliamo gli indici sintetici T^2,
100 # statistica: T2(m,n) = T2_i
101 T2.stat = apply(Z, 1, function(z){
102     sum(z^2 / lambda[1:n.pc])})
103
104 # Applichiamo il k-means clustering
105 # alle statistiche T^2
106 cat("\nSTPCA: kmeans con 2 cluster")
107 K2 = kmeans(T2.stat, centers = 2)
108
109 # Otteniamo le statistiche SSW2
110 # e le salviamo
111
112 SSW2[(frame.index - n_frame_start + 1)] = K2$tot.withinss
113
114 # Aggiornamento del frame j: frame.index
115 frame.index = frame.index + 1
116 }
117 return(SSW2)
118 }
119
120
121 # Funzione per il calcolo delle Statistiche EWMA
122 # in input:
123 # - ssw: statistiche ssw(2)
```

```
124 #         calcolate con la funzione ssw2.fun
125 # - lambda: parametro della ewma (0.6 di default)
126
127 ewma.fun = function(ssw, lambda = 0.6){
128   EWMA = numeric(length(ssw))
129   for(i in 1:length(EWMA)){
130     if (i == 1) EWMA[i] = ssw[i]
131     else EWMA[i] = (1-lambda)*EWMA[i-1] + lambda*(ssw[i])
132   }
133   return(EWMA)
134 }
135
136
137
138 # Funzioni per il calcolo delle Statistiche WCuSum
139
140 # 1) Funzione per il calcolo dei pesi EWMA:
141 # i pesi ewma vengono calcolati per le
142 # statistiche che vanno da 2 a 122,
143 # quindi gli indici della EWMA sono shiftati.
144 # Ritorna i residui e i pesi.
145 # in INPUT:
146 # - ssw: statistiche ssw(2)
147 #         calcolate con la funzione ssw2.fun
148 # - lambda: parametro della ewma (0.6 di default)
149
150 pesi.fun = function(ssw, lambda = 0.6){
151   yt = ssw[-1]                # ssw meno la prima osservazione
152   yprec = ssw[-length(ssw)]   # ssw ritardata di una osservazione
153   fit = lm(yt ~ yprec)
154   res = yt - predict(fit, as.data.frame(yprec))
155   PESI = numeric(length(res))
156   for(i in 1:length(PESI)){
157     if (i == 1) {PESI[i] = res[i]}
158     else {PESI[i] = (1-lambda)*PESI[i-1] + lambda*(res[i])}
159   }
```

```
160 out.list = list(res = res, pesi = PESI)
161 return(out.list)
162 }
163
164 # 2) Funzione per il calcolo delle statistiche WcusSum:
165 # Calcola i pesi e i residui
166 # richiamando la funzione pesi.fun
167 # Assegna 0 come valore iniziale per i residui e per i pesi
168 # in INPUT:
169 # - ssw: statistiche ssw2 calcolate con la funzione ssw2.fun
170 # - lambda: parametro da assegnare al peso EWMA
171 # - k: parametro della statistica WcusSum
172
173 wcus.fun = function(ssw, lambda = 0.6, k)
174 {
175   out.list = pesi.fun(ssw, lambda)
176   # i residui e pesi saranno da (1, ..., 122)
177   # --> primi valori posti a 0
178   res = c(0, out.list$res)
179   pesi = c(0, out.list$pesi)
180   wcus = numeric(122)
181   for(i in 1:122){
182     if(i == 1) {wcus[i] = res[i]}
183     else {wcus[i] = max(0, wcus[i-1] + abs(pesi[i])*(res[i] - k))}
184   }
185   return(wcus)
186 }
```

B.4 Calcolo dei limiti di controllo

Di seguito viene riportata la funzione `bisez.fun` utilizzata per il calcolo dei limiti di controllo, e i relativi comandi utilizzati per ottenere (i) le matrici di statistiche in controllo: `sims.ssw`, `sims.ewma` e `sims.wcus` (Appendice A), e (ii) i valori critici, riportati in Tabella A.1.

La funzione `bisez.fun` implementa il metodo di bisezione descritto in Se-

zione 4.2.2. Prende in input una tra le matrici delle statistiche in controllo: `sims.ssw`, `sims.ewma` e `sims.wcus`, il valore target A_0 dell'ARL_{IC} (ARL) che si vuole garantire e il numero massimo di iterazioni (`maxiter`), e ritorna il limite superiore h richiesto.

```

1 # Funzione per il calcolo del limite unilaterale
2 # superiore h tramite il metodo di bisezione
3 # in INPUT:
4 # - sims: matrice 500 x 122 delle statistiche
5 #       in controllo, ogni riga relativa
6 #       alle statistiche calcolate
7 #       sull'i-esimo video, con i= 1, ..., 500.
8 # - ARL: valore dell'ARL_IC che si vuole garantire.
9 # - maxiter: numero massimo di iterazioni prima di
10 #          terminare l'algoritmo (20 di default)
11
12 bisez.fun <- function(sims, ARL, maxiter = 20) {
13
14   # Inizializzazione dei valori
15
16   hmax_bi <- max(sims) # h_U
17   hmin_bi <- 0.0      # h_L
18   B <- nrow(sims)     # numero di video da ricampionare
19   maxrl <- ncol(sims) # numero di statistiche per ogni
20                       # video = massima Run Length
21
22   RLs <- numeric(B)  # Conterra' le B Run Length
23                       # osservate ad ogni iterazione
24
25   hold <- 1.0         # Valore iniziale di ARL_IC
26   i <- 0              # indice delle iterazioni
27   h <- 0.0           # Valore iniziale del limite h
28
29   # Statistiche relative al video che sar
30   # campionato di volta in volta
31   # e sul quale sara' calcolata la RL

```

```
32  sims_boot <- numeric(maxrl)
33
34  # Inizio del metodo di bisezione:
35  # iterazioni per trovare h
36
37  while(i < maxiter) {
38    cat("\n", i)
39    i <- i + 1
40    h <- (hmin_bi + hmax_bi) / 2
41
42    # Ad ogni iterazione campiono
43    # casualmente B righe (B video)
44
45    for(b in 1:B) {
46
47      # Campiono casualmente un video e
48      # su questo calcolo la Run Length,
49      # utilizzando come limite
50
51      sims_boot <- sims[sample(1:B, 1),]
52      for(n in 1:maxrl) {
53        if((sims_boot[n] > h) | (n == maxrl)){
54          RLS[b] <- n
55          break
56        }
57      }
58    }
59
60    # Calcolo l'ARL_ic osservato come
61    # la media delle RL
62
63    E_RL <- mean(RLs)
64
65    # Aggiornamneto dei parametri
66    if (E_RL > ARL) {
67      hmax_bi <- h
```

```
68   } else if (E_RL < ARL) {
69     hmin_bi <- h
70   }
71
72   # Criterio per la convergenza dell'algoritmo
73   if (abs(hold - h) < 0.001) {
74     break
75   }
76   # Nuovo ARL_IC ottenuto
77   hold <- h
78 }
79 return(h)
80 }
81
82
83 # Calcolo dei limiti di controllo
84 # utilizzando la funzione bisez.fun
85 # La lista SSW2.list contiene tutte le
86 # statistiche ssw2 calcolate sui video
87 # generati con il bootstrap a blocchi
88
89 # Le statistiche di controllo EWMA e WCUSUM
90 # dei vengono attenute nel seguente modo,
91 # e organizzate nelle matrici di dimensione
92 # 500 x 122: sims.ssw, sims.ewma e sims.wcus.
93 # Queste statistiche entrano quindi in
94 # input nella funzione bisez.fun
95
96 sims.ssw = sims.ewma = sims.wcus = matrix(0, nrow = 500, ncol = 122)
97 for(i in 1:500){
98   sims.ssw[i,] = SSW2.list[[i]]
99   sims.ewma[i,] = ewma.fun(sims.ssw[i,], lambda = 0.6)
100  sims.wcus[i,] = wcus.fun(sims.ssw[i,], lambda = 0.6, k = 0.01)
101 }
102
103 # Calcolo dei limiti per tre valori di ARL
```



```
104 # che si vogliono garantire: ARL_IC = {30, 60, 90}
105 set.seed(0000)
106 ARL_IC1 = 30 # -----
107 L1_ssw = find_h(sims.ssw, ARL = ARL_IC1)
108 L1_ewma = find_h(sims.ewma, ARL = ARL_IC1)
109 L1_wcus = find_h(sims.wcus, ARL = ARL_IC1)
110
111
112 ARL_IC2 = 60 # -----
113 L2_ssw = find_h(sims.ssw, ARL = ARL_IC2)
114 L2_ewma = find_h(sims.ewma, ARL = ARL_IC2)
115 L2_wcus = find_h(sims.wcus, ARL = ARL_IC2)
116
117
118 ARL_IC3 = 90 # -----
119 L3_ssw = find_h(sims.ssw, ARL = ARL_IC3)
120 L3_ewma = find_h(sims.ewma, ARL = ARL_IC3)
121 L3_wcus = find_h(sims.wcus, ARL = ARL_IC3)
```

B.5 Calcolo delle ARL_{IC} osservate

Di seguito viene riportato il codice utilizzato per il calcolo delle \widehat{ARL}_{IC} osservate e riportate in Tabella 4.1.

```
1 # I limiti calcolati con il metodo di bisezione (funzione bisez.fun),
2 # per i diversi valori di ARL_IC vengono confrontati con le
3 # 500 statistiche calcolate sui video in controllo generati con il
4 # bootstrap a blocchi, le quali sono organizzate negli elementi:
5 # sims.ssw, sims.ewma e sims.wcus. Quando la carta non chiama allarme
6 # la RL_IC viene posta pari a 122, ovvero all'indice massimo delle
7 # statistiche rilevate. L'ARL_IC viene calcolato come
8 # la media delle RL_IC.
9
10 # Gli elementi L1..., L2..., L3...
11 # sono i limiti di controllo per le 3
```

```
12 # ARL_IC relative ad ogni carta di controllo
13
14 # SSW -----
15 set.seed(000)
16 RL1.ssw.IC = apply(sims.ssw, 1, function(x) min(which(x>L1_ssw)))
17 RL1.ssw.IC[RL1.ssw.IC == Inf] = 122
18 mean(RL1.ssw.IC)
19 sd(RL1.ssw.IC)
20
21 RL2.ssw.IC = apply(sims.ssw, 1, function(x) min(which(x>L2_ssw)))
22 RL2.ssw.IC[RL2.ssw.IC == Inf] = 122
23 mean(RL2.ssw.IC)
24 sd(RL2.ssw.IC)
25
26 RL3.ssw.IC = apply(sims.ssw, 1, function(x) min(which(x>L3_ssw)))
27 RL3.ssw.IC[t3_allarme.ssw.IC == Inf] = 122
28 mean(RL3.ssw.IC)
29 sd(RL3.ssw.IC)
30
31 # EWMA -----
32 set.seed(000)
33 RL1.ewma.IC = apply(sims.ewma, 1, function(x) min(which(x>L1_ewma)))
34 RL1.ewma.IC[RL1.ewma.IC == Inf] = 122
35 mean(RL1.ewma.IC)
36 sd(RL1.ewma.IC)
37
38 RL2.ewma.IC = apply(sims.ewma, 1, function(x) min(which(x>L2_ewma)))
39 RL2.ewma.IC[RL2.ewma.IC == Inf] = 122
40 mean(RL2.ewma.IC)
41 sd(RL2.ewma.IC)
42
43 RL3.ewma.IC = apply(sims.ewma, 1, function(x) min(which(x>L3_ewma)))
44 RL3.ewma.IC[RL3.ewma.IC == Inf] = 122
45 mean(RL3.ewma.IC)
46 sd(RL3.ewma.IC)
47
```

```
48 # WCUSUM -----
49 set.seed(000)
50 RL1.wcus.IC = apply(sims.wcus, 1, function(x) min(which(x>L1_wcus)))
51 RL1.wcus.IC[RL1.wcus.IC == Inf] = 122
52 mean(RL1.wcus.IC)
53 sd(RL1.wcus.IC)
54
55 RL2.wcus.IC = apply(sims.wcus, 1, function(x) min(which(x>L2_wcus)))
56 RL2.wcus.IC[RL2.wcus.IC == Inf] = 122
57 mean(RL2.wcus.IC)
58 sd(RL2.wcus.IC)
59
60 RL3.wcus.IC = apply(sims.wcus, 1, function(x) min(which(x>L3_wcus)))
61 RL3.wcus.IC[RL3.wcus.IC == Inf] = 122
62 mean(RL3.wcus.IC)
63 sd(RL3.wcus.IC)
```

B.6 Simulazione di anomalie termiche

Di seguito vengono riportate le funzioni implementate per la simulazione di anomalie termiche. In particolare, `hotspot.fun` genera, sul video dato in input, un'anomalia di dimensione e durata desiderabili, a partire dal fotogramma j_0 . Tale funzione è stata modificata e tradotta dal codice Matlab originale, utilizzato nelle simulazioni di Colosimo e Grasso (2018).

Le funzioni `xy.fun` e `location.fun` servono, rispettivamente, per l'indicizzazione dei p pixel e per stabilire da quali fotogrammi l'anomalia può essere simulata nell' i -esimo video. Entrambe le funzioni vengono richiamate in `hotspot.fun`.

La funzione `hotspot.fun`, in accordo con quanto descritto nella Sezione 5.1, permette quindi di simulare, su qualsiasi video, un'anomalia desiderata, dando in input: (i) i dati video (`dati`) in formato *folded*, (ii) la dimensione ν dell'anomalia che si vuole generare (`hot_size`, ci sono 6 possibilità), (iii) la durata ω dell'anomalia (`hot_duration`), (iv) il fotogramma dal quale si desidera fare insorgere l'anomalia (`hot_start`). Inoltre con il parametro `new` è

possibile generare l'anomalia sia utilizzando la (5.1), sia le Formule 5.2 e 5.3. La funzione necessita infine le posizioni del laser rilevate nel corso dei fotogrammi del video originale (`dataIC`), le quali sono contenute nella matrice `location` (si veda Appendice A).

La funzione ritorna in output il nuovo video simulato in formato *array*.

Di seguito sono riportati i comandi utilizzati per la generazione dei video fuori controllo, i cui scenari sono descritti nella Sezione 5.2. In particolare, tali video sono ottenuti utilizzando la funzione `hotspot.fun`, dove sono stati utilizzati come parametri i valori (4, 5) per `hot_size`, relativi rispettivamente alle dimensioni $\nu = \{45, 80\}$, e (30, 50) per `hot_duration`, mentre i relativi valori per `hot_start` sono campionati casualmente e salvati nella lista `Index.HS` (si veda Appendice A).

I video simulati non sono salvati, vengono invece calcolate direttamente le statistiche $SSW(2)$, utilizzando la funzione `ssw2.fun` (appendice B.3), sui nuovi video simulati, le quali vengono salvate nella lista `SSW2.OC` (si veda Appendice A).

```
1 # Funzione che crea la matrice di indicizzazione dei pixel
2 # il pixel di coordinate (m,n) viene indicizzato in d,
3 # con d = 1, ..., p
4 # in INPUT:
5 # - l'array del video
6
7 xy.fun = function(dati)
8 {
9   DIMx = dim(dati)[1]
10  DIMy = dim(dati)[2]
11  Nframes = dim(dati)[3]
12  k <- 1
13  XY <- matrix(nrow = DIMx * DIMy, ncol = 2)
14  for (j in 1:DIMy) {
15    for (i in 1:DIMx) {
16      XY[k, ] <- c(i, j)
17      k <- k + 1
18    }
19  }
```

```
19 }
20 return(XY)
21 }
22
23
24
25 # Funzione che crea la matrice delle posizioni del laser
26 # per ogni video dato
27 # in INPUT:
28 # - start_location: matrice delle locazioni del video
29 #                   originale
30 # - ind: indice del video per cui si vogliono trovare
31 #       le posizioni ind = 1, ..., 500
32 #       (0 di default: video originale)
33 # - Index.video: indici dei frames del video originale
34 #               di cui e' composto il video dato in
35 #               input
36 # - RETURN: TRUE: ritorna la matrice delle posizioni
37 # - possible.starting.hot: TRUE: mostra gli indici dei
38 #                           frames a partire dai
39 #                           quali non possono essere
40 #                           simulati gli hot spot.
41 # Gli indici che non appartengono all'intervallo (10-110)
42 # vengono posti pari ad 1, e non si potra' far partire
43 # la simulazione degli hot spot da quei frames
44
45 location.fun = function(start_location = start_location,
46                        Index.video = Index.video, ind = 0,
47                        RETURN = T, possible.starting.hot = F)
48 {
49 # Se video originale --> la posizione del laser e'
50 # quella contenuta in start_location
51 if(ind == 0) {return(start_location)}
52
53 # Salvo gli indici dei frames del video ind-esimo
54 ind.i = Index.video[[ind]]
```

```
55
56 # Pongo pari a 1 i frames minori di 11 e maggiori
57 # di 110: da questi non posso generare
58 ind.i[ind.i<11 | ind.i >110] = 1
59
60 # Definiscono la nuova matrice di location
61 loc.i = start_location[ind.i,]
62 row.names(loc.i) = 1:161
63 cat("\nGli hot_spot non possono essere simulati a partire dai seguenti
64     frames:\n\n",
65     which(ind.i == 1))
66 if(possible.starting.hot){
67     return(which(ind.i != 1))
68 }
69 if(RETURN) {return(loc.i)}
70 }
71
72 # Funzione per la simulazione di hot spot:
73 # ritorna il video in formato array (folded)
74 # con hot spot
75 # in INPUT:
76 # - dati: l'array relativo al video in controllo
77 #         su cui si vuole simulare l'anomalia
78 # - hot_size: dimensione/forma dell' HS
79 #             (6 possibilita', ma si possono
80 #             introdurre piu' alternative);
81 # - hot_duration: durata (in frame) dell' HS;
82 # - hot_start: frame in cui inizia l'HS:
83 #             questo ha luogo nel centroide del laser;
84 # - start_location: matrice delle posizioni del laser
85 #                 del video originale
86 # - ind: indice del video per cui si vogliono trovare le posizioni
87 #       ind = 1, ..., 500 (0 se video originale)
88 # - H: definisce il profilo di raffreddamento dell' HS;
89 # - new: determina il metodo di simulazione: TRUE:
```

```
90 #         metodo proposto nell'elaborato,
91 #         FALSE: metodo proposto da Colosimo e Grasso (2018).
92
93 hotspot.fun = function(dati, hot_size, hot_duration, hot_start,
94                       H = 0.95, start_location = start_location, ind,
95                       Index.video = Index.video.IC, new = T)
96 {
97   cat("\nViene restituito un video con Hot Spot delle stesse dimensioni
98       del video dato in in input")
99   cat("\nHot spot Simulation ... \n")
100  cat("\nHot Spot size:", hot_size)
101  cat("\nHot Spot duration:", hot_duration)
102  cat("\nHot Spot start at frame:", hot_start, "\n")
103
104  # Indice del frame in cui terminera' l'anomalia
105  hot_end = hot_start + hot_duration
106  # Dimensione dei dati originali
107  DIMx = dim(dati)[1]
108  DIMy = dim(dati)[2]
109  # Posizioni del laser quando inizia l' HS
110  cat("\n\nUnfolding del video...")
111  data.unf = unfolding.fun(dati)
112  cat("\n\nCalcolo della matrice di location del laser...")
113  loc.i = location.fun(start_location = start_location,
114                      Index.video = Index.video, ind = ind)
115  # Indicizzo i pixel con la funzione xy.fun
116  # e salvo i pixel centroidi dell'anomalia
117  XY = xy.fun(dati)
118  YY <- loc.i[hot_start, 1]
119  XX <- loc.i[hot_start, 2]
120
121  # Creazione della matrice che conterra' il
122  # video (unfolded) con l'anomalia
123  MATT <- data.unf
124
125  # Salvo i pixel interessati dall' HS
```

```
125 id_hot1 <- NULL
126
127 # Scelta del tipo di HS
128 switch (hot_size,
129       {
130       # size = 4 (square shape)
131       id_hot1[1:2] <- which(XY[, 1] == XX & XY[, 2] >= YY - 1 &
132                          XY[, 2] < YY + 1)
133       id_hot1[3:4] <- which(XY[, 1] == XX - 1 & XY[, 2] >= YY - 1 &
134                          XY[, 2] < YY + 1)
135     },
136     {
137     # size = 9 (square shape)
138     id_hot1[1:3] <- which(XY[, 1] == XX & XY[, 2] > YY - 2 & XY[,
139                          2] < YY + 2)
140     id_hot1[4:6] <- which(XY[, 1] == XX - 1 & XY[, 2] > YY - 2 &
141                          XY[, 2] < YY + 2)
142     id_hot1[7:9] <- which(XY[, 1] == XX + 1 & XY[, 2] > YY - 2 &
143                          XY[, 2] < YY + 2)
144   },
145   {
146   # size = 20 (cross-shaped)
147   id_hot1[1] <- which(XY[, 1] == XX & XY[, 2] == YY)
148   id_hot1[2] <- which(XY[, 1] == XX & XY[, 2] == YY + 1)
149   id_hot1[3] <- which(XY[, 1] == XX + 1 & XY[, 2] == YY)
150   id_hot1[4] <- which(XY[, 1] == XX + 1 & XY[, 2] == YY + 1)
151   id_hot1[5] <- which(XY[, 1] == XX & XY[, 2] == YY - 2)
152   id_hot1[6] <- which(XY[, 1] == XX & XY[, 2] == YY - 1)
153   id_hot1[7] <- which(XY[, 1] == XX & XY[, 2] == YY + 2)
154   id_hot1[8] <- which(XY[, 1] == XX & XY[, 2] == YY + 3)
155   id_hot1[9] <- which(XY[, 1] == XX + 1 & XY[, 2] == YY - 2)
156   id_hot1[10] <- which(XY[, 1] == XX + 1 & XY[, 2] == YY - 1)
157   id_hot1[11] <- which(XY[, 1] == XX + 1 & XY[, 2] == YY + 2)
158   id_hot1[12] <- which(XY[, 1] == XX + 1 & XY[, 2] == YY + 3)
159   id_hot1[13] <- which(XY[, 1] == XX - 1 & XY[, 2] == YY)
160   id_hot1[14] <- which(XY[, 1] == XX - 1 & XY[, 2] == YY + 1)
```



```
156     id_hot1[15] <- which(XY[, 1] == XX - 2 & XY[, 2] == YY)
157     id_hot1[16] <- which(XY[, 1] == XX - 2 & XY[, 2] == YY + 1)
158     id_hot1[17] <- which(XY[, 1] == XX + 2 & XY[, 2] == YY)
159     id_hot1[18] <- which(XY[, 1] == XX + 2 & XY[, 2] == YY + 1)
160     id_hot1[19] <- which(XY[, 1] == XX + 3 & XY[, 2] == YY)
161     id_hot1[20] <- which(XY[, 1] == XX + 3 & XY[, 2] == YY + 1)
162 },
163 {
164     # size = 45 (cross-shaped)
165     id_hot1[1:9] <- which(XY[, 1] > XX - 2 & XY[, 1] < XX + 2 &
166         XY[, 2] > YY - 5 & XY[, 2] < YY - 1)
167     id_hot1[10:18] <- which(XY[, 1] > XX - 2 & XY[, 1] < XX + 2 &
168         XY[, 2] > YY + 1 & XY[, 2] < YY + 5)
169     id_hot1[19:27] <- which(XY[, 1] > XX - 5 & XY[, 1] < XX - 1 &
170         XY[, 2] > YY - 2 & XY[, 2] < YY + 2)
171     id_hot1[28:36] <- which(XY[, 1] > XX + 1 & XY[, 1] < XX + 5 &
172         XY[, 2] > YY - 2 & XY[, 2] < YY + 2)
173     id_hot1[37:45] <- which(XY[, 1] > XX - 2 & XY[, 1] < XX + 2 &
174         XY[, 2] > YY - 2 & XY[, 2] < YY + 2)
175 },
176 {
177     # size = 80 (cross-shaped)
178     id_hot1[1:16] <- which(XY[, 1] > XX - 3 & XY[, 1] < XX + 2 &
179         XY[, 2] > YY - 7 & XY[, 2] < YY - 2)
180     id_hot1[17:32] <- which(XY[, 1] > XX - 3 & XY[, 1] < XX + 2 &
181         XY[, 2] > YY + 1 & XY[, 2] < YY + 6)
182     id_hot1[33:48] <- which(XY[, 1] > XX - 7 & XY[, 1] < XX - 2 &
183         XY[, 2] > YY - 3 & XY[, 2] < YY + 2)
184     id_hot1[49:64] <- which(XY[, 1] > XX + 1 & XY[, 1] < XX + 6 &
185         XY[, 2] > YY - 3 & XY[, 2] < YY + 2)
186     id_hot1[65:80] <- which(XY[, 1] > XX - 3 & XY[, 1] < XX + 2 &
187         XY[, 2] > YY - 3 & XY[, 2] < YY + 2)
188 },
189 {
190     # size = 100 (square shape)
191     id_hot1[1:100] <- which(XY[,1] >= XX - 5 & XY[,1] <= XX + 4 &
```

```
182             XY[,2] >= YY - 5 & XY[,2] <= YY + 4)
183         }
184     )
185     cat("\n\nPixel con hot spot: ", id_hot1, "\n\n")
186
187     # Simulazione con metodo proposto in (Sezione 5.1)
188     if(new){
189
190         # Funzione che definisce il raffreddamento
191         # dei pixel interessati dall' HS
192         y <- 1/(1 + exp(0.2 * (1:hot_duration - H * hot_duration)))
193
194         # Nuova luminosita' dei pixel interessati dall'HS,
195         # viene tolto 80 (luminosita' media di background)
196         C <- y * 255 - 80
197         for(id in 1:length(id_hot1)){
198             for(d in 1: hot_duration){
199
200                 # Aggiorno la luminosita' dei pixel interessati
201                 # dall'anomalia
202                 MATT[id_hot1[id], hot_start + d] <- min(255, MATT[id_hot1[id],
203                     hot_start + d] + C[d] )
204             }
205         }
206
207         # Simulazione con metodo proposto da
208         # Colosimo e Grasso (2018)
209
210     }else{
211
212         # Funzione che definisce il raffreddamento
213         # dei pixel interessati dall'HS
214
215         y <- 1/(1 + exp(0.2 * (1:hot_duration - H * hot_duration)))
216         # Nuova luminosita' dei pixel interessati dall'HS
```

```

217   C <- y * 255
218   for(id in 1:length(id_hot1)){
219     for(d in 1: hot_duration){
220
221       # Aggiorno la luminosita' dei pixel interessati
222       # dall'anomalia
223       MATT[id_hot1[id], hot_start + d] <- C[d]
224     }
225   }
226 }
227
228 # IM: video (folded) fuori controllo (OC)
229 IM = folding.fun(MATT, DIMx = DIMx, DIMy = DIMy)
230
231 return(IM)
232 }

```

```

1 set.seed(1234)
2 # ANOMALIA 1: {n = 45, t = 30} -----
3 # Generiamo gli indici j_0 da cui simuleremo
4 # l'anomalia A1
5 HS.Real.Medium.30_starts = sample(11:110, 30)
6 # Inizializziamo la lista che conterra' le
7 # statistiche SSW2
8 SSW.Real.Medium.30 = list()
9
10 k = 0
11
12 # Genero video fuori controllo partendo da
13 # HS.Real.Medium.30_starts,
14 # compresi tra 11 e 110
15
16 for(t in HS.Real.Medium.30_starts){
17   k = k + 1
18   cat("\n\nIterazione:", k, " \ \ \ Indice del frame:", t, "\n")
19   video.hs = hot.spot(dati = dataic, hot_size = 4, hot_duration = 30,

```

```

20         hot_start = HS.Real.Medium.30_starts[k], ind = 0,
21         start_location = start_location, Index.video =
           Index.video.IC)
22 video.hs.unf = unfolding.fun(video.hs)
23 SSW.Real.Medium.30[[k]] = ssw2.fun(video.hs.unf, W.mat = W.mat)
24 }
25
26
27 # ANOMALIA 2: {n = 45, t = 50} () -----
28 # Generiamo gli indici j_0 da cui simuleremo
29 # l'anomalia A2
30 HS.Real.Medium.50_starts = sample(11:110, 30)
31 # Inizializziamo la lista che conterra' le
32 # statistiche SSW2
33 SSW.Real.Medium.50 = list()
34
35 k = 0
36
37 # Genero video fuori controllo partendo da
38 # HS.Real.Medium.50_starts,
39 # compresi tra 11 e 110
40
41 for(t in HS.Real.Medium.50_starts){
42   k = k + 1
43   cat("\n\nIterazione:", k, " \ \ \ Indice del frame:", k, "\n")
44   video.hs = hot.spot(dati = dataic, hot_size = 4, hot_duration = 50,
45                       hot_start = HS.Real.Medium.50_starts[k], ind = 0,
46                       start_location = start_location, Index.video =
           Index.video.IC)
47   video.hs.unf = unfolding.fun(video.hs)
48   SSW.Real.Medium.50[[k]] = ssw2.fun(video.hs.unf, W.mat = W.mat)
49 }
50
51 # ANOMALIA 3: {n = 80, t = 30} () -----
52 # Generiamo gli indici j_0 da cui simuleremo
53 # l'anomalia A3

```

```
54 HS.Real.BIG.30_starts = sample(11:110, 30)
55 # Inizializziamo la lista che conterra' le
56 # statistiche SSW2
57 SSW.Real.BIG.30 = list()
58
59 k = 0
60
61 # Genero video fuori controllo partendo da
62 # HS.Real.BIG.30_starts,
63 # compresi tra 11 e 110
64
65 for(t in HS.Real.BIG.30_starts){
66   k = k + 1
67   cat("\n\nIterazione:", k, " \ \ \ Indice del frame:", k, "\n")
68   video.hs = hot.spot(dati = dataic, hot_size = 5, hot_duration = 30,
69                       hot_start = HS.Real.BIG.30_starts[k], ind = 0,
70                       start_location = start_location, Index.video =
71                           Index.video.IC)
72   video.hs.unf = unfolding.fun(video.hs)
73   SSW.Real.BIG.30[[k]] = ssw2.fun(video.hs.unf, W.mat = W.mat)
74 }
75 # ANOMALIA 4: {n = 80, t = 50} () -----
76 # Generiamo gli indici j_0 da cui simuleremo
77 # l'anomalia A4
78 HS.Real.BIG.50_starts = sample(11:110, 30)
79 # Inizializziamo la lista che conterra' le
80 # statistiche SSW2
81 SSW.Real.BIG.50 = list()
82
83 k = 0
84
85 # Genero video fuori controllo partendo da
86 # HS.Real.BIG.50_starts,
87 # compresi tra 11 e 110
88
```

```

89 for(t in HS.Real.BIG.50_starts){
90   k = k + 1
91   cat("\n\nIterazione:", k, " \ \ \ Indice del frame:", k, "\n")
92   video.hs = hot.spot(dati = dataic, hot_size = 5, hot_duration = 50,
93                     hot_start = HS.Real.BIG.50_starts[k], ind = 0,
94                     start_location = start_location, Index.video =
95                               Index.video.IC)
96   video.hs.unf = unfolding.fun(video.hs)
97   SSW.Real.BIG.50[[k]] = ssw2.fun(video.hs.unf, W.mat = W.mat)
98 }
99
100 # Salviamo gli oggetti SSW2.0C, e Index.HS
101 SSW2.0C = list(ssw2.A1 = SSW.Real.Medium.30,
102              ssw2.A2 = SSW.Real.Medium.50,
103              ssw2.A3 = SSW.Real.BIG.30,
104              ssw2.A4 = SSW.Real.BIG.50)
105 Index.HS = list(index.A1 = HS.Real.Medium.30_starts,
106               index.A2 = HS.Real.Medium.50_starts,
107               index.A3 = HS.Real.BIG.30_starts,
108               index.A4 = HS.Real.BIG.50_starts)

```

B.7 Calcolo delle ARL_{OC}

Le ARL_{OC} , riportate nella Tabella 5.2. sono state ottenute, in accordo alla (2.3), con i seguenti comandi. Si riporta il solo codice relativo al calcolo dell' ARL_{OC} sui video simulati con la prima anomalia (scenario A_1), e per la statistica $SSW(2)$, in quanto i comandi sono analoghi per gli altri campioni simulati, e per EWMA e $WCuSum$. In particolare, vengono inizialmente calcolate le statistiche EWMA e $WCuSum$, a partire dalle statistiche $SSW(2)$ salvate nella lista `SSW.0C` (Appendice A), utilizzando le funzioni `ewma.fun` e `wcus.fun` (Appendice B.3). Vengono quindi calcolati gli istanti in cui viene segnalato l'allarme e le relative ARL_{OC} e deviazioni standard.

```

1 # Calcolo delle statistiche di controllo

```

```
2 # con le funzioni ewma.fun e wcus.fun
3 ssw.oc1 = ewma.oc1 = wcus.oc1 = matrix(0, nrow = 30, ncol = 122)
4 for(i in 1:30){
5   ssw.oc1[i,] = SSW.OC[[1]]
6   ewma.oc1[i,] = ewma.fun(ssw.oc1[i,], lambda = 0.6)
7   wcus.oc1[i,] = wcus.fun(ssw.oc1[i,], lambda = 0.6, k = 0.01)
8 }
9
10 # ---> SSW
11
12 # Salviamo il primo momento t in cui viene chiamato allarme
13 # ovvero in cui la statistica eccede il rispettivo limite
14 t1_allarme.ssw.oc1 = apply(ssw.oc1, 1, function(x) min(which(x>L1_ssw)))
15 t2_allarme.ssw.oc1 = apply(ssw.oc1, 1, function(x) min(which(x>L2_ssw)))
16 t3_allarme.ssw.oc1 = apply(ssw.oc1, 1, function(x) min(which(x>L3_ssw)))
17
18 # Calcoliamo le RL, sottraendo al tempo di allarme il
19 # frame j_0 nel quale e' insorta l'anomalia e aggiungendo
20 # i 40 frame di batch, in questo modo otteniamo il
21 # numero di fotogrammi necessari per segnalare l'allarme
22 # quando e' insorta una anomalia
23
24 RL1_ssw.oc1 = t1_allarme.ssw.oc1 + 40 - Index.HS[[1]]
25 RL1_ssw.oc1[RL1_ssw.oc1 == Inf] = NA
26 mean(RL1_ssw.oc1, na.rm = TRUE)
27 sd(RL1_ssw.oc1, na.rm = TRUE)
28
29 RL2_ssw.oc1 = t2_allarme.ssw.oc1 + 40 - Index.HS[[1]]
30 RL2_ssw.oc1[RL2_ssw.oc1 == Inf] = NA
31 mean(RL2_ssw.oc1, na.rm = TRUE)
32 sd(RL2_ssw.oc1, na.rm = TRUE)
33
34 RL3_ssw.oc1 = t3_allarme.ssw.oc1 + 40 - Index.HS[[1]]
35 RL3_ssw.oc1[RL3_ssw.oc1 == Inf] = NA
36 mean(RL3_ssw.oc1, na.rm = TRUE)
37 sd(RL3_ssw.oc1, na.rm = TRUE)
```


Bibliografia

- Armingol, J. M. et al. (2003). «Statistical pattern modeling in vision-based quality control systems». In: *Journal of Intelligent and Robotic Systems* 37, pp. 321–336.
- Bugatti, M. e Colosimo, B. M. (2022). «Towards real-time in-situ monitoring of hot-spot defects in L-PBF: a new classification-based method for fast video-imaging data analysis». In: *Journal of Intelligent Manufacturing* 33.1, pp. 293–309.
- Bühlmann, P. (2002). «Bootstraps for time series». In: *Statistical science*, pp. 52–72.
- Chen, C. H. (2015). *Handbook of pattern recognition and computer vision*. World scientific.
- Chen, S. e Wells, L. (2022). «A multi-image monitoring framework for statistical process control to improve manufacturing systems». In: *Quality and Reliability Engineering International* 38.2, pp. 679–702.
- Colosimo, B. M. e Grasso, M. (2018). «Spatially weighted PCA for monitoring video image data with application to additive manufacturing». In: *Journal of Quality Technology* 50.4, pp. 391–417.
- Colosimo, B. M. et al. (2018). «Opportunities and challenges of quality engineering for additive manufacturing». In: *Journal of Quality Technology* 50.3, pp. 233–252.
- Dong, C.-Z. e Catbas, F. N. (2021). «A review of computer vision-based structural health monitoring at local and global levels». In: *Structural Health Monitoring* 20.2, pp. 692–743.
- Efron, B. e Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

- Eslami, D. et al. (2023). «Statistical monitoring of image data using multi-channel functional principal component analysis». In: *Communications in Statistics-Theory and Methods* 52.12, pp. 4165–4182.
- Gibson, I. et al. (2021). *Additive manufacturing technologies*. Vol. 17. Springer.
- Grasso, M et al. (2021). «In-situ measurement and monitoring methods for metal powder bed fusion: an updated review». In: *Measurement Science and Technology* 32.11.
- Grasso, M. e Colosimo, B. M. (2017). «Process defects and in situ monitoring methods in metal powder bed fusion: a review». In: *Measurement Science and Technology* 28.4.
- Grasso, M. et al. (2017). «In-process monitoring of selective laser melting: spatial detection of defects via image data analysis». In: *Journal of Manufacturing Science and Engineering* 139.5.
- Hastie T., R. T. e Friedman, J. (2001). *The elements of statistical learning*. New York: Springer.
- He, Z. et al. (2016). «An image-based multivariate generalized likelihood ratio control chart for detecting and diagnosing multiple faults in manufactured products». In: *International Journal of Production Research* 54.6, pp. 1771–1784.
- Jolliffe, I. T. (2002). *Principal component analysis for special types of data*. Springer.
- Lu, H., Plataniotis, K. N. e Venetsanopoulos, A. N. (2008). «MPCA: Multilinear Principal Component Analysis of Tensor Objects». In: *IEEE Transactions on Neural Networks* 19.1, pp. 18–39.
- Malik, K. et al. (2023). «Computer vision models for comparing spatial patterns: understanding spatial scale». In: *International Journal of Geographical Information Science* 37.1, pp. 1–35.
- Megahed, F. M. et al. (2012). «A spatiotemporal method for the monitoring of image data». In: *Quality and Reliability Engineering International* 28.8, pp. 967–980.
- Park, C. e Shrivastava, A. K. (2014). «Multimode geometric-profile monitoring with correlated image data and its application to nanoparticle self-

- assembly processes». In: *Journal of Quality Technology* 46.3, pp. 216–233.
- Patton, A., Politis, D. N. e White, H. (2009). «Correction to “Automatic block-length selection for the dependent bootstrap” by D. Politis and H. White». In: *Econometric Reviews* 28.4, pp. 372–375.
- Pham, D. T. e Gault, R. S. (1998). «A comparison of rapid prototyping technologies». In: *International Journal of machine tools and manufacture* 38.10-11, pp. 1257–1287.
- Politis, D. N. e White, H. (2004). «Automatic block-length selection for the dependent bootstrap». In: *Econometric reviews* 23.1, pp. 53–70.
- Qiu, P. (2013). *Introduction to statistical process control*. CRC press.
- Qiu, P. (2020). «Big data? Statistical process control can help!» In: *The American Statistician* 74.4, pp. 329–344.
- Russ, J. e Neal, F. (2016). *The image processing handbook*. London: Taylor & Francis Group.
- Shu, L., Jiang, W. e Tsui, K.-L. (2008). «A weighted CUSUM chart for detecting patterned mean shifts». In: *Journal of Quality Technology* 40.2, pp. 194–213.
- Tong, L.-I., Wang, C.-H. e Huang, C.-L. (2005). «Monitoring defects in IC fabrication using a Hotelling T/sup 2/control chart». In: *IEEE transactions on semiconductor manufacturing* 18.1, pp. 140–147.
- Yan, H., Paynabar, K. e Shi, J. (2014). «Image-based process monitoring using low-rank tensor decomposition». In: *IEEE Transactions on Automation Science and Engineering* 12.1, pp. 216–227.
- Yang, W. et al. (2023). «A tensor-based hierarchical process monitoring approach for anomaly detection in additive manufacturing». In: *Quality and Reliability Engineering International* 39.2, pp. 630–650.
- Zuo, L., He, Z. e Zhang, M. (2020). «An EWMA and region growing based control chart for monitoring image data». In: *Quality Technology & Quantitative Management* 17.4, pp. 470–485.