

POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria



**ALGORITMO DI TAMPERING
DETECTION OTTIMIZZATO
TRAMITE SEGMENTAZIONE DELLA
SCENA INQUADRATA**

Relatore: Prof. Giacomo BORACCHI
Correlatore: Ing. Claudio MARCHISIO

Tesi di Laurea di:
Adriano GAIBOTTI, matricola 780200

Anno Accademico 2013-2014

A Sara

Sommario

Uno dei principali problemi, quando si ha a che fare con applicazioni di monitoraggio video, è quello di identificare quegli eventi che possano compromettere la corretta ripresa della scena da parte del sensore. Può capitare, ad esempio, che dell'acqua piovana si depositi sulla lente della camera, rendendo l'immagine acquisita sfocata, oppure che qualcuno sposti la camera in modo che essa non riprenda più la scena che stava monitorando. Il problema di individuare, in maniera automatica, questo tipo di eventi prende il nome di *tampering detection*. Nella letteratura scientifica lo studio di questo problema si è concentrato solamente sulle applicazioni di *videosorveglianza*, dove è necessario che la camera operi con una frequenza di acquisizione delle immagini elevata. Lo scopo della tesi è lo sviluppo di un algoritmo di *tampering detection* per sistemi di monitoraggio a basso consumo. Abbiamo considerato scenari, come ad esempio il monitoraggio ambientale, in cui non è necessario che la camera operi con un'acquisizione continua e, quindi, sia possibile limitare il carico computazionale acquisendo, ad esempio, un'immagine ogni minuto. In questi casi, se consideriamo il caso della ripresa di una strada, in cui passano delle macchine o dei pedoni, abbiamo un'elevata dinamicità che non permette di fare un confronto tra frame consecutivi per identificare gli eventi di nostro interesse. In aggiunta, abbiamo dei cambiamenti di luminosità, tra un'immagine e la successiva, più sostanziali rispetto al caso di acquisizione continua. La nostra proposta è quella di monitorare nel tempo degli indicatori semplici, calcolati considerando solamente il *contenuto visivo* delle singole immagini, in modo da ottimizzare le risorse hardware disponibili e ridurre la complessità computazionale. Data l'alta variabilità di questi indicatori abbiamo introdotto una *segmentazione* della scena ripresa, estratta durante una fase di *configurazione* dell'algoritmo, in modo da considerare solo le regioni in cui il monitoraggio risulta più efficace. La tesi è stata svolta durante uno stage presso *STMicroelectronics*, particolarmente interessata a sviluppare algoritmi intelligenti di processione immagini da integrare nei propri dispositivi hardware, e a scenari di impiego per questi. Prove sperimentali, fatte considerando diversi sistemi di acquisizione operanti con frame rate differenti, hanno confermato l'efficacia di utilizzare la segmentazione rispetto a considerare l'intera scena per individuare eventi di spostamento della camera.

Ringraziamenti

Ringrazio

Indice

Sommario	iii
Ringraziamenti	v
1 Introduzione	3
2 Stato dell'arte	5
2.1 Tampering Detection	5
2.1.1 Identificazione di occlusioni	5
2.1.2 Identificazione di spostamenti della camera	5
2.1.3 Identificazione di sfocature	5
3 Impostazione del problema di ricerca	7
3.1 Modello delle osservazioni	7
3.1.1 Sfocatura	7
3.1.2 Spostamento della camera	9
3.1.3 Occlusione	10
3.2 Tampering detection	10
4 Soluzione proposta	11
4.1 Estrazione dei descrittori del cambiamento	11
4.2 Algoritmo di segmentazione	11
4.3 Monitoraggio one-shot	11
4.4 Monitoraggio sequenziale	11
5 Realizzazioni sperimentali e valutazione	13
5.1 Acquisizione dei dataset	13
5.2 Risultati	13
6 Direzioni future di ricerca e conclusioni	15
Bibliografia	17

Elenco delle figure

3.1	Esempi di sfocature	8
3.2	Sequenza di otto frame consecutivi acquisiti ogni minuto . . .	9
3.3	Esempio di spostamento della camera	10

Elenco delle tabelle

Capitolo 1

Introduzione

Negli ultimi anni le applicazioni di tipo multimediale sono aumentate in maniera esponenziale, soprattutto per quanto riguarda i contenuti video. L'abbassamento dei prezzi e delle dimensioni dei *sensori* e delle componenti hardware

Capitolo 2

Stato dell'arte

In questo capitolo elenchiamo quelle che sono le principali tecniche, presenti nella letteratura scientifica, utilizzate per identificare tentativi di manomissione su camere di videosorveglianza.

2.1 Tampering Detection

Nei moderni sistemi di videosorveglianza troviamo spesso algoritmi utilizzati per identificare particolari eventi all'interno della scena ripresa dalla camera. Ad esempio è possibile avere un software in grado di identificare le targhe delle automobili che superano il limite di velocità , oppure la presenza di oggetti incustoditi in una stazione [1]. Affinché questi algoritmi funzionino correttamente, è importante che le immagini, che verranno poi processati da questi sistemi, mantengano una certa qualità.

2.1.1 Identificazione di occlusioni

2.1.2 Identificazione di spostamenti della camera

2.1.3 Identificazione di sfocature

Capitolo 3

Impostazione del problema di ricerca

In questo capitolo descriviamo il problema, affrontato dall'algoritmo di tampering detection, in maniera formale e rigorosa. Il primo paragrafo illustra il modello delle osservazioni e gli eventi che siamo interessati a identificare, mentre il secondo paragrafo formalizza il concetto di tampering detection. Nel seguito useremo il termine *scena* per indicare l'ambiente ripreso dalla camera, determinato univocamente dalla posizione della stessa¹.

3.1 Modello delle osservazioni

Il nostro campo di osservazione si concentra su quegli eventi che si interpongono tra la scena ripresa da una camera e il sensore che acquisisce le immagini. Non vogliamo, cioè, identificare degli eventi particolari che avvengono nella scena, come un oggetto lasciato incustodito [1], bensì vogliamo identificare quegli eventi tali per cui il sensore non è più nelle condizioni di riprendere, in maniera ottimale, la scena, quali ad esempio sfocature o spostamenti della camera. Nel seguito cerchiamo di dare una definizione formale di questi eventi.

3.1.1 Sfocatura

Il fenomeno della sfocatura avviene quando un elemento trasparente o semi-trasparente si interpone tra la lente della camera e la scena ripresa, oppure quando viene cambiata la messa a fuoco, causando una perdita nei dettagli della scena ripresa.

Nella figura 3.1 sono mostrati degli esempi in cui sono presenti delle sfocature. Queste possono essere di origine diversa:

¹Altre volte useremo il termine *inquadratura*.

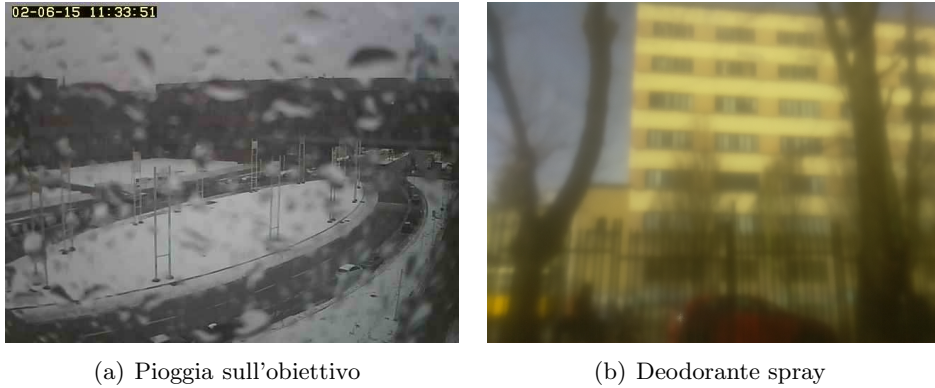


Figura 3.1: Esempi di sfocature

- dovute a *cause naturali*, come ad esempio dell'acqua piovana che si deposita sulla lente (figura 3.1(a)), o la condensa dovuta all'umidità e alle basse temperature, oppure un raggio di sole incidente sull'obiettivo della camera;
- per *intervento dell'uomo*, che a sua volta può avvenire in maniera intenzionale (e in questo caso si può parlare di *manomissione*) oppure non intenzionale. Ad esempio, si può direttamente intervenire sulla messa a fuoco, nel caso sia possibile cambiarla manualmente; oppure (come nel caso della figura 3.1(b)) è possibile applicare una sostanza semitrasparente sulla lente della camera, come il gas di un deodorante spray.

Riprendendo [2], questo fenomeno può essere modellato come un operatore di *degradazione* \mathcal{D} applicato a un'immagine y , considerata priva di errori, i.e.,

$$z = \mathcal{D}[y]. \quad (3.1)$$

In particolare, all'interno dell'operatore \mathcal{D} si può considerare il contributo dovuto a un operatore di *sfocatura* \mathcal{B} (dall'inglese *blur*) e un termine aleatorio η corrispondente al rumore, i.e.,

$$z(x) = \mathcal{D}[y](x) = \mathcal{B}[y](x) + \eta(x), \quad x \in \mathcal{X} \quad (3.2)$$

dove abbiamo indicato con x le coordinate dei *pixel* dell'immagine e \mathcal{X} è l'insieme dei pixel che formano l'immagine. Considerando il caso continuo, possiamo assumere la sfocatura \mathcal{B} come un operatore *lineare*,

$$\mathcal{B}[y](x) = \int_{\mathcal{X}} y(s)h(x, s)ds, \quad (3.3)$$

dove $h(x, s)$ rappresenta la *risposta all'impulso* (*point spread function* (PSF)) della sfocatura sul pixel x , il cui risultato consiste nel rendere le differenze di intensità, tra pixel adiacenti, più morbide (*smooth*). Nel caso in cui



Figura 3.2: Sequenza di otto frame consecutivi acquisiti ogni minuto

la sfocatura sia applicata sulla totalità dell'immagine (come nel caso della figura 3.1(b)), allora è possibile modellare l'operatore di blur come una *convoluzione*²:

$$\mathcal{B}[y](x) = \int_{\mathcal{X}} y(s)h(s-x)ds, \quad (3.4)$$

dove $h(\cdot)$ è un filtro gaussiano o uniforme.

Nel caso più generale possiamo considerare che la camera acquisisca un sequenza di N osservazioni $\{z_i\}, i = 1, \dots, N$, quindi la formula 3.2 si può riscrivere come

$$z_i(x) = \mathcal{D}_i[y_i](x) = \mathcal{B}_i[y_i](x) + \eta(x), \quad x \in \mathcal{X}. \quad (3.5)$$

La sequenza delle immagini $\{y_i\}, i = 1, \dots, N$, può variare in maniera significativa nel suo contenuto, anche nel caso in cui la scena ripresa sia la stessa. Un esempio è illustrato nella figura 3.2, in cui le immagini riprese dalla camera sono acquisite ogni minuto. Nonostante la scena ripresa sia la stessa, il suo contenuto delle singole immagini varia parecchio, dato il continuo passaggio di automobili e pedoni. Questo problema fa sì che l'identificazione delle sfocature non possa avvenire facendo un semplice confronto tra frame consecutivi, in quanto avremmo un numero troppo elevato di falsi positivi. Infatti, nel caso in cui avessimo un riscontro negativo dal confronto tra due frame ($z_i \neq z_{i+1}$), sarebbe difficile capire se è cambiato il contenuto delle immagini ($y_i \neq y_{i+1}$) o l'operatore di sfocatura ($\mathcal{B}_i \neq \mathcal{B}_{i+1}$).

3.1.2 Spostamento della camera

Lo spostamento della camera avviene quando cambia la scena ripresa. Le cause possono essere, ancora una volta, di tipo naturale, ad esempio una raffica di vento che sposta la camera, oppure dovute a un intervento malevolo da parte di un uomo. Un esempio di spostamento della camera è mostrato nella figura 3.3. Possiamo formalizzare il concetto di spostamento

²Il blur convoluzionale è quello che abbiamo utilizzato per generare, in maniera sintetica, sequenze con immagini sfocate.



Figura 3.3: Esempio di spostamento della camera

della camera nel modo seguente: consideriamo la sequenza $\{y_i\}$ di immagini generate da una camera in una certa posizione, e la sequenza $\{w_i\}$ di immagini generate dalla stessa camera in una posizione differente.

Possiamo, dunque, considerare la sequenza di immagini $\{z_i\}$ in cui avviene uno spostamento della camera all'istante T^* nel seguente modo:

$$z_i(x) = \begin{cases} y_i(x) + \eta(x) & \text{per } i < T^* \\ w_i(x) + \eta(x) & \text{per } i \geq T^*, \end{cases} \quad (3.6)$$

dove $\eta(x)$ è un rumore stazionario.

Anche per lo spostamento della camera vale la considerazione fatta nel caso della sfocatura: il contenuto delle immagini varia con il passare del tempo, quindi identificare lo spostamento confrontando frame consecutivi nel tempo genera un alto numero di falsi positivi.

3.1.3 Occlusione

3.2 Tampering detection

Capitolo 4

Soluzione proposta

4.1 Estrazione dei descrittori del cambiamento

4.2 Algoritmo di segmentazione

4.3 Monitoraggio one-shot

4.4 Monitoraggio sequenziale

Capitolo 5

Realizzazioni sperimentali e valutazione

5.1 Acquisizione dei dataset

5.2 Risultati

Capitolo 6

Direzioni future di ricerca e conclusioni

Bibliografia

- [1] <http://www.mitan.it/security-solution/videosorveglianza/sistemi-di-videosorveglianza-e-registrazione/>. Visitato il giorno 09/03/2015.
- [2] Cesare Alippi, Giacomo Boracchi, Romolo Camplani, and Manuel Roveri. Detecting external disturbances on the camera lens in wireless multimedia sensor networks. *Instrumentation and Measurement, IEEE Transactions on*, 59(11):2982–2990, 2010.