



**PhD in Information Technology and Electrical Engineering**

**Università degli Studi di Napoli Federico II**

**PhD Student: Luca D'Amiano**

---

**XXX Cycle**

**Training and Research Activities Report – Second Year**

**Tutor: prof. Giovanni Poggi**



# Training and Research Activities Report – Second Year

PhD in Information Technology and Electrical Engineering – XXX Cycle

Luca D’Amiano

## 1. Information

Luca D’Amiano, MSc in Ingegneria delle Telecomunicazioni – Università di Napoli Federico II

XXX Cycle- ITEE – Università di Napoli Federico II

Tutor: Giovanni Poggi

## 2. Study and Training activities

### a. Courses

- i. Ad hoc Course, “Game Theory and analysis of competitive dynamics for industrial systems”, Gennaio 2016 (3 cfu)
- ii. Ad hoc Course, “Complementi di Analisi Funzionale”, Aprile 2016 (6 cfu) (da completare)
- iii. English course at CLA, April 2016 (40 Ore)

### b. Seminars

- i. “La sintesi sonora dell’ingegnere Laurens Hammond, QUANDO LA SCIENZA INCONTRA L’ARTE” - Riccardo de Asmundis – 9 November 2015 (4 ore)
- ii. Model Based and Pattern Based GUI Testing – Prof. Ana Paiva - University of Oporto – 23/25 November 2015 (0.8 CFU)
- iii. Armi autonome, problemi etici e decisioni politiche – Prof. G. Tamburrini – 1 December 2015 (0.2 CFU)
- iv. Perception-Based Surround Sound Recording and Reproduction - Dr. Enzo De Sena - 22 February 2016 (1 ora)
- v. An overview on image forensics with emphasis on physics-based scene verification - Dr. Christian Riess – 18 May 2016 (0.2 CFU)

### c. External courses

- i. ICVSS 2016, International Computer Vision Summer School, July 2016 (3 CFU)
- ii. IEEE EURASIP S3P 2016, Summer School on Signal Processing, September 2016 (3 CFU)

	Credits year 1		Credits year 2								Credits year 3		Total	Check
	Estimated	Summary	Estimated	bimonth 1	bimonth 2	bimonth 3	bimonth 4	bimonth 5	bimonth 6	Summary	Estimated	Summary		
<b>Modules</b>	18	19	9		3	4		3		10	12	12	41	30-70
<b>Seminars</b>	13	5,2	6	1,8	0,2		0,2		3	5,2	6	6	16,4	10-30
<b>Research</b>	34	36	42	7	7	7	7	7	7	42	45	45	123	80-140
	65	60,2	57	8,8	10,2	11	7,2	10	10	57,2	63	63	180,4	180

3. Research activity

**Copy move Video forgery detection**

Durante questo secondo anno di dottorato mi sono occupato di video forgery detection estendendo la tecnica presentata l’anno scorso e migliorandone le performance.

La tecnica di base che è stata estesa si compone di tre parti principali:

- estrazione delle features
- matching
- post processing

Lo scopo dell’algoritmo è quello di individuare regioni duplicate in un video. Per fare ciò si calcola il campo di nearest-neighbor dell’immagine (NNF) tramite la tecnica di matching e con il post processing si va ad individuare se ci sono zone dell’NNF lineari, ovvero che indicano matching uniformi di intere aree del video. In realtà, per rendere l’algoritmo più robusto, il matching non viene effettuato nel dominio del video, ma in un dominio trasformato; questo dominio trasformato si ottiene tramite il calcolo di features sulla griglia completa dei pixel.

Questa tecnica di base è stata utilizzata per sviluppare un algoritmo più performante in termini di complessità computazionale. In figura 1 è riportato lo schema a blocchi dell’algoritmo proposto.

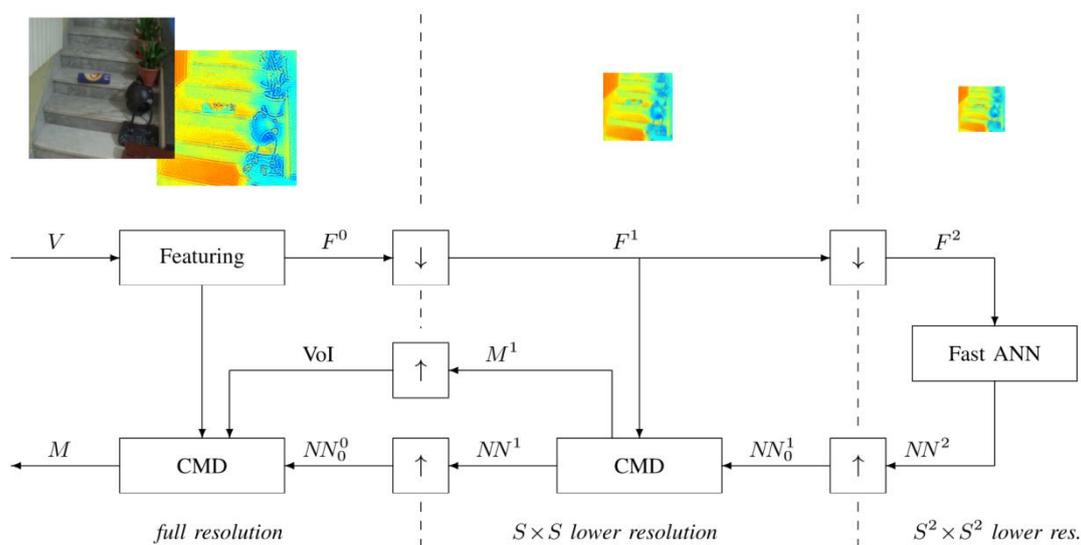


Figura 1: schema a blocchi dell’algoritmo proposto

L’algoritmo presenta una struttura a multi-risoluzione.

Il primo passo dell’algoritmo consiste nell’estrazione delle features (questo passo sarà analizzato in dettaglio nella prossima sezione). A valle dell’estrazione delle features si effettuano due sottocampionamenti del video nel dominio delle features e si applica la tecnica di base descritta precedentemente (riquadro CMD dello schema a blocchi). In particolare, per il video a più bassa risoluzione, si utilizza una tecnica di matching differente rispetto alle altre risoluzioni. Questa tecnica è più lenta ma più accurata.

Il campo di nearest-neighbor in uscita al livello più basso di risoluzione viene sovracampionato alla risoluzione intermedia e va ad inizializzare l’algoritmo di mathing (PatchMatch) di questo livello. La maschera binaria in uscita a questo livello individua dei volumi di interesse (VOI). L’ultimo passo dell’algoritmo consiste

nell’applicare la tecnica di base solo ai volumi di interesse individuati al passo precedente. In figura 2 è riportato lo pseudo-codice dell’algoritmo descritto.

---

**Algorithm** Multi-resolution Video Copy-Move Detector

---

<b>Require:</b> $V$	▷ input video
<b>Ensure:</b> $M$	▷ output detection map
1: $F^0 = \text{FeatureExtract}(V)$	▷ will work on features from now on
2: $F^1 = F^0 \downarrow S$	▷ $S \times S$ downsampling
3: $F^2 = F^1 \downarrow S$	▷ $S \times S$ downsampling
4: $NN^2 = \text{FastNNSearch}(F^2)$	▷ conventional NN search at level 0
5: $NN_0^1 = NN^2 \uparrow S$	▷ initial estimate of $NN^1$
6: $[M^1, NN^1] = \text{CMD}(F^1, NN_0^1)$	▷ CMD at level 1
7: $M_0^0 = M^1 \uparrow S$	▷ $M_0^0$ gives the VoI
8: $NN_0^0 = NN^1 \uparrow S$	▷ initial estimate of $NN^0$
9: $[M, NN^0] = \text{CMD}(F^0, NN_0^0, \text{VoI})$	▷ CMD at level 0 on VoI

---

Figura 2: pseudo-codice dell’algoritmo proposto

Nelle seguenti sezioni sono descritti alcuni dettagli della tecnica appena descritta.

**Estrazione delle Features**

La prima fase dell’algoritmo consiste nel calcolare i descrittori (features) sulla griglia completa dei pixel del video (sliding mode). Si è scelto di utilizzare il valore assoluto della trasformata di Fourier su griglia polare adottando come profilo radiale i momenti di Zernike.

$$f(n, m) = \int_0^\infty \rho R_{n,m}^*(\rho) \times \left[ \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} I(\rho, \theta) e^{-jm\theta} d\theta \right] d\rho$$

$$R_{n,m}(\rho) = \begin{cases} \sqrt{c_n} \sum_{h=0}^{\frac{n-|m|}{2}} \frac{(-1)^h (n-h)!}{h! (\frac{n+|m|}{2}-h)! (\frac{n-|m|}{2}-h)!} \rho^{n-2h}, & \rho \leq 1 \\ 0, & otherwise \end{cases}$$

$n - |m|$  is nonnegative and even

Queste features risultano essere robuste rispetto a distorsioni quali compressione e rumore additivo, inoltre sono invarianti per rotazione. Al fine di rendere l’algoritmo più robusto rispetto ai falsi allarmi (causati dal matching indesiderato di piccole regioni del video), le features così calcolate, frame per frame, vengono concatenate secondo la seguente equazione:

$$g(t) = \begin{cases} \frac{1}{\sqrt{2}} |f(t) + f(-t)| & t > 0 \\ |f(t)| & t = 0 \\ \frac{1}{\sqrt{2}} |f(t) - f(-t)| & t < 0 \end{cases}$$

Queste features risultano essere invarianti per flip-temporali.

## 4. Products

### Conference

D'Amiano, L., Cozzolino, D., Poggi G., & Verdoliva, L. (2015, June). "Video forgery detection and localization based on 3D patchmatch". Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on (pp.1-6) IEEE.

### Journal

D'Amiano, L., Cozzolino, D., Poggi G., & Verdoliva, L. "A PatchMatch-based Dense-field Algorithm for Video Copy-Move Detection and Localization". IEEE Transactions on Circuits and Systems for Video Technology (submitted)

## 5. Conferences and Seminars

-

## 6. Activity abroad

-

## 7. Tutorship

-