
ELABORATO FINALE DELL'ATTIVITÀ SVOLTA PRESSO IL TIROCINIO

Laboratorio Segnali e Immagini, Istituto di Scienza e Tecnologia
dell'Informazione "A. Faedo" CNR-ISTI

Tirocinante: Maria Laura Cuttano (mat. 604414, Università di Pisa,
Dipartimento di Ingegneria Civile e Industriale, Laurea Triennale in Ingegneria
Gestionale)

Periodo attività tirocinio: 15/05/2025 – 21/07/2025

Numero di ore: 200

Codice di riferimento del progetto formativo: 028338

Codice di riferimento della convenzione: 013846

Tutor aziendali: Davide Moroni, Giulio Del Corso

Relatore della tesi: Leonardo Marrazzini



UNIVERSITÀ DI PISA

**DIPARTIMENTO DI INGEGNERIA CIVILE E INDUSTRIALE
LAUREA TRIENNALE IN INGEGNERIA GESTIONALE**

ELABORATO PER LA PROVA FINALE

***Progettazione di Codici 2D
per la Tracciabilità Circolare di Cavi***

RELATORI

Prof. Ing. Leonardo Marrazzini

Dipartimento di Ingegneria Civile e Industriale

IL CANDIDATO

Maria Laura Cuttano

m.cuttano1@studenti.unipi.it

23/07/2025

1. INTRODUZIONE E PROGETTO	3
2. SISTEMI DI CODIFICA 2D PIÙ DIFFUSI E ALGORITMO DI REED-SOLOMON	3
2.1 CODICI 2D ESISTENTI	3
2.2 ALGORITMO DI ENCODING E DECODING DI REED-SOLOMON	4
3. SVILUPPO DI UNA MARCATURA ORIGINALE	5
3.1 ESIGENZE E VINCOLI DELL'AZIENDA.....	5
3.2 PARAMETRI INPUT.....	5
3.3 STEP DELL'ALGORITMO DI COSTRUZIONE DEL CODICE	6
3.3.1 CALCOLO DELLA RIDONDANZA MASSIMA PERMESSA DATI I PARAMETRI IN INPUT	6
3.3.2 CONVERSIONE ED ENCODING TRAMITE ALGORITMO DI REED-SOLOMON.....	6
3.3.3 CREAZIONE MATRICE MARKER.....	6
3.3.4 DISPOSIZIONE STRINGA CODIFICATA NELLA MATRICE MARKER	7
4. CONCLUSIONI	7
BIBLIOGRAFIA E SITOGRAFIA.....	8

1. Introduzione e progetto

Questa tesi descrive l'attività di tirocinio svolta presso il "Laboratorio Segnali e Immagini" dell'Istituto di Scienza e Tecnologia dell'Informazione "A. Faedo" CNR-ISTI su "CircularEconomyCable", progetto condotto dall'azienda produttrice di cavi "Tratos S.p.A."

Il progetto, della durata prevista di due anni, mira allo sviluppo e prototipazione di innovativi cavi elettrici e in fibra ottica ad elevata sostenibilità, tracciabilità e visibilità, per consentire la piena attuazione dei principi dell'economia circolare e della transizione digitale nei settori del trasporto energia e della connettività dati.

Con questo scopo, oltre agli obiettivi in ambito chimico come lo sviluppo di polimeri sostenibili per la produzione delle guaine dei cavi (bio-based, da riciclo) o il miglioramento delle loro funzionalità in termini di visibilità e resistenza tramite additivi innovativi, il progetto ha anche richiesto in ambito software lo sviluppo di una marcatura originale da imprimere sulla guaina dei cavi, direttamente in fase di estrusione, per registrare, tracciare e certificare in modo sicuro la vita del cavo, dall'installazione, alla manutenzione, allo smaltimento.

Nell'arco del tirocinio, ispirandomi alla letteratura preesistente sui codici bidimensionali, ho sviluppato e implementato un package Python capace di rispondere alle esigenze dell'azienda nel generare marker univoci e ridondanti adatti all'elevata variabilità dei cavi da produrre.

2. Sistemi di codifica 2D più diffusi e algoritmo di Reed-Solomon

2.1 Codici 2D esistenti

I codici bidimensionali rappresentano un metodo, ormai di larga diffusione, per contenere grandi quantità di informazioni in uno spazio ridotto. Attraverso strumenti di lettura, spesso riducibili a semplici fotocamere, i dati al loro interno sono recuperati e decodificati in modo rapido e affidabile.



Figura 1: "Codici bidimensionali più diffusi"

Le molteplici tipologie già esistenti (esempi in Fig. 1) si differenziano per forma, velocità di lettura, dimensione dell'informazione codificata e campo di applicazione (es: il PDF417 è usato soprattutto sui biglietti, mentre la Data Matrix su piccoli componenti elettronici), ma condividono degli elementi comuni:

- **Unità fondamentale:** la singola unità di informazione, che può essere il “pixel” (bianco o nero a seconda che rappresenti uno 0 o un 1 in binario) per il QR Code, lo spessore della linea nei codici a barre o il “puntino” nei codici MaxiCode.
- **Marker di posizione:** segnalano la presenza del codice in fase di lettura e forniscono un riferimento di scala e di eventuali distorsioni per permettere una corretta visualizzazione e decodifica (es: i “quadrati” agli angoli del QR Code, le barre ai lati dei PDF417, le barre nere e a “trattini” che incorniciano le Data Matrix e così via).
- **Sistemi di ridondanza:** tecniche utilizzate per rendere i codici più resistenti agli errori. Introducono delle informazioni aggiuntive nel codice, in modo che questo possa ancora essere letto correttamente anche se parzialmente danneggiato, sporco o coperto. La maggior parte dei sistemi di ridondanza (compreso quello adottato in questo progetto) si basano sulla “Codifica di Reed-Solomon”.

2.2 Algoritmo di encoding e decoding di Reed-Solomon

Sviluppato nel 1960 da Irving Reed e Gustave Solomon, è il metodo di correzione degli errori alla base di innumerevoli sistemi di memorizzazione e trasmissione dati, come i codici QR, decodificabili anche se con parti mancanti, distorti o rovinati (Fig. 2) e i CD leggibili anche se graffiati.



Figura 2: Codici QR "rovinati" ma leggibili

L’algoritmo aggiunge al messaggio che si intende memorizzare o trasmettere (memorizzato su una quantità “m” di simboli) un numero “k” di simboli di correzione (calcolati con polinomiali su campi finiti, spesso usando come base algebrica 2^8 , quindi l’unità base di informazione nel nostro codice 2D ai fini di codifica e decodifica non sarà il “pixel” del singolo bit ma un “gruppo di 8 pixel consecutivi”, ovvero un byte).

Tramite questi simboli di correzione aggiunti al messaggio iniziale l’algoritmo è in grado di correggere, in qualsiasi parte del codice, fino a:

- $k/2$ errori se non conosco la loro posizione.
- k errori se conosco la loro posizione.

Fra le numerose implementazioni efficienti dell’algoritmo di Reed-Solomon, ho sfruttato quella resa disponibile dal package “*reedsolo*”, così da concentrarmi sulla codifica grafica necessaria a questa particolare applicazione.

3. Sviluppo di una marcatura originale

Partendo dalle esigenze dell'azienda, sono stati definiti i parametri necessari alla generazione del codice finale (matrice rettangolare di pixel bianchi e neri), a sua volta ottenuto tramite step di conversione, encoding, generazione dei marker e tassellamento della matrice.

3.1 Esigenze e vincoli dell'azienda

- **ID cavo:** registrare univocamente una produzione di circa 10'000 cavi l'anno, che stimata una vita utile di 20-30 anni prima dello smaltimento, e considerato un cospicuo margine di sicurezza, porta ad un upper-target di inventario di 1'000'000 di tipi di cavi diversi, indicabile con un identificativo a stringa binaria di 20 cifre.

- **Informazione sulla distanza tra due punti del cavo:** ripetizione periodica e regolare del codice ad una frequenza stabilita, mantenendo però una differenza su pixel scelti "di metraggio" per poter, in seguito alla scannerizzazione di codici in punti diversi, calcolare la distanza tra questi (ad esempio per sapere lo sviluppo in lunghezza di cavi interrati).

- **Vincolo sulla dimensione:** a seconda del diametro del cavo è possibile permettersi marcature dalle dimensioni ("pixel in altezza" x "pixel in larghezza") differenti, i cavi prodotti da Tratos S.p.A., a seconda dello scopo, variano da un diametro di 10mm per gli Overhead Power Ground Wire a un diametro di 120mm per cavi HV interrati. A causa dell'escursione metrica di inventario dobbiamo quindi creare dei codici con parametro libero sulle dimensioni e fare in modo che il codice possa essere non solo quadrato ma anche rettangolare.

- **Vincolo per processo realizzativo:** i codici 2D devono essere stampati, direttamente in fase di estrusione della guaina del cavo, da una macchina imprimitrice, la quale necessariamente porrà dei vincoli su:

- **risoluzione** (dimensione minima del singolo "pixel").
- **geometria** (ad esempio sarà impossibile realizzare un codice anulare)

3.2 Parametri input

Sulla base delle esigenze dell'azienda è stato scelto di non adottare un codice già esistente ma di crearlo da zero rendendolo il più flessibile possibile per meglio adattarsi alla dimensione del cavo e all'ambiente in cui opera. I parametri lasciati "liberi" e quindi da fornire da caso a caso (oltre all'ID del cavo) sono:

- **Numero di righe e colonne** (dipendenti dal diametro del cavo).

- **Tipologia, dimensione e numero di marker di posizione:** similmente ai codici QR e alle Data Matrix è possibile inserire fino a quattro marker quadrati (agli angoli) o a linea (sui lati), modificabili in dimensione e numero, ottimizzando lo spazio a disposizione.
- **Frequenza delle ripetizioni per il “metraggio”:** più è alta la frequenza di ripetizione e maggiore sarà il numero di bit su cui indicare l’informazione.

3.3 Step dell’algoritmo di costruzione del codice

Per spiegare le fasi di costruzione del codice finale seguiamo lo sviluppo a partire dai parametri di input per un esempio di codice di dimensioni pari a 15x22, con 2 marker quadrati 6x6, che rappresenti l’ID “10101011011” situato alla ripetizione sullo stesso cavo numero 179 (in binario “10110011”, che, ipotizzando 2 codici al metro, indicherebbe il metro numero 90).

3.3.1 Calcolo della ridondanza massima permessa dati i parametri in input

Dato il numero di righe e colonne, l’ID, il tipo/dimensione dei marker e il numero di bit da allocare al metraggio l’algoritmo calcola per differenza il numero massimo di bit che è possibile allocare alla ridondanza:

$$\text{bit ridondanza} = (\text{righe} * \text{colonne}) - \text{bit messaggio} - \text{bit marker} - \text{bit metraggio}$$

3.3.2 Conversione ed encoding tramite algoritmo di Reed-Solomon

- Prende in input l’ID del cavo in binario (eg: "10101011011"), se il numero di cifre (bit) dell’ID non è un multiplo di 8 applica un’operazione di “padding”, ovvero inserisce in cima un numero di zeri tale da renderlo multiplo di 8 (eg: “0000010101011011”).
- Divide la stringa a gruppetti da 8 bit e li converte in elementi byte (eg: “00000101” -> “\x05”, “01011011” -> “[”).
- Applica l’encoding del pacchetto “reedsolo” aggiungendo un numero k di elementi di ridondanza (calcolati allo step precedente) ai simboli del messaggio iniziale.
- Riconverte gli elementi in binario per poterli inserire nella matrice (punto 4.3.4).

3.3.3 Creazione matrice marker

Sulla base dei parametri forniti viene creata la matrice delle dimensioni desiderate, su cui vengono applicati i marker (in Fig. 3 esempi di marker possibili, in Fig. 4 il nostro esempio).



Figura 3: "Esempi di marker creabili"

3.3.4 Disposizione stringa codificata nella matrice marker

Il danneggiamento recuperabile dall'algoritmo di Reed-Solomon è considerato sull'elemento in byte (8 bit successivi) e non sul singolo bit, è necessario perciò disporre la stringa all'interno della matrice in modo che gli elementi dei "gruppi di bit" (evidenziati in Fig.4 con colori diversi successivi) siano compatti, è stata quindi adottata una tassellatura con una funzione che percorre la matrice a serpentina verticale, alternando il senso di lettura da sinistra a destra e viceversa in caso incontri "ostacoli" (bordi, marker o pixel già inseriti dalla tassellatura).



Figura 4: A) Tassellatura a serpentina, B) Disposizione stringa codificata, C) Aggiunta bit metraggio

- I primi bit (Fig. 4B, in azzurro) sono inizialmente ignorati per la generazione del codice poiché dedicati all'informazione "di metraggio", diversi per due punti distinti dello stesso cavo (sono calcolati direttamente all'imprimitura tramite contatore, es: Fig. 4C "10110011").

- I bit restanti sono occupati dalla stringa codificata.

Fig. 4C rappresenta il codice stampato nel punto e sul cavo descritti dai parametri.

Successivamente in fase di lettura verranno: identificati i marker, creata una suddivisione virtuale a griglia per individuare i singoli pixel (correggendo eventuali curvature), riconosciuto il valore binario dei pixel, decodificata la stringa letta, e ottenuta l'informazione di partenza.

4. Conclusioni

Implementare un algoritmo di correzione degli errori come Reed-Solomon per costruire da zero una marcatura consente di avere maggiore flessibilità e controllo rispetto ai codici preesistenti e il package Python da me sviluppato (che, una volta ultimato e rifinito, sarà caricato su GitHub), abbinato all'inventario dell'azienda, rappresenta un esempio pratico. Inoltre, a queste tipologie di codici possono essere affiancate tecniche di Computer Vision e Machine Learning in grado di potenziare ulteriormente la correzione degli errori. Tali tecniche, "allenandosi" su grandi dataset di immagini (vere o generate) di codici danneggiati o coperti, possono riuscire a "ripulire" parzialmente l'immagine, garantendo maggiore robustezza all'algoritmo e aumentando la capacità di recupero delle informazioni.

Bibliografia e sitografia

Tratos Cavi S.p.A. (s.d.). *Tipologie di cavi prodotti*. Recuperato da [https://tratosgroup.com/about-us/facilities/tratos-italy-pieve-santo-stefano-cable-factory/T24 Economia+2Tratos Group+2Tratos Group+2](https://tratosgroup.com/about-us/facilities/tratos-italy-pieve-santo-stefano-cable-factory/T24%20Economia+2Tratos%20Group+2Tratos%20Group+2)

Masabi. (2011). *Connecting the Dots: An Introduction to 2D Barcodes*. Recuperato da [https://www.masabi.com/2011/03/04/connecting-the-dots-an-introduction-to-2d-barcodes-3/ reddit.com+13masabi.com+13blog.masabi.com+13](https://www.masabi.com/2011/03/04/connecting-the-dots-an-introduction-to-2d-barcodes-3/)

Denso Wave Inc., *QR Code Essentials*, Denso Wave, 1994. [Online]. Recuperato da: <https://www.qrcode.com/en/>

Hackaday. (2016, 10 febbraio). *Error Detection and Correction: Reed–Solomon, Convolution And Trellis Diagrams*. Recuperato da <https://hackaday.com/2016/02/10/error-detection-and-correction-reed-solomon-convolution-and-trellis-diagrams/>

DataGenetics. (s.d.). *QR Code Resilience*. Recuperato da [http://datagenetics.com/blog/march42013/ \(datagenetics.com\)](http://datagenetics.com/blog/march42013/)

Filiba, T. (2023). *reedsolo (versione 1.7.0)*. Recuperato da <https://pypi.org/project/reedsolo/>

Latha, Y. M., & Rao, B. S. (2023). *Advanced Denoising Model for QR Code Images Using Hough Transformation and Convolutional Neural Networks*. *Traitement du Signal*, 40(3), 1243–1249. Recuperato da <https://doi.org/10.18280/ts.400342> ijeta.org+1mendeley.com+1

Maria Laura Cuttano. (2025). *AdversarialRedundant2DCode* [Repository GitHub]. Recuperato da <https://github.com/GDelCorso/AdversarialRedundant2DCode>