

Comunicazioni Multimediali

Progetto CV 1:

Tracking vehicles in a parking lot: interface devel.

Vico Mantoan, Giulio Maria Facin

Dipartimento di ingegneria e scienze dell'informazione

Indice

1. [Descrizione Progetto:](#)
2. [Interfaccia:](#)
 - 2.1 [Modalità Admin:](#)
 - 2.2 [Modalità Utente:](#)
 - 2.3 [Altro:](#)
3. [Spiegazione File XML:](#)
4. [Autoadattamento:](#)
 - 4.1 [Primo approccio:](#)
 - 4.1.1 [Problemi:](#)
 - 4.2 [Approccio definitivo:](#)
 - 4.3 [Descrizione algoritmo:](#)

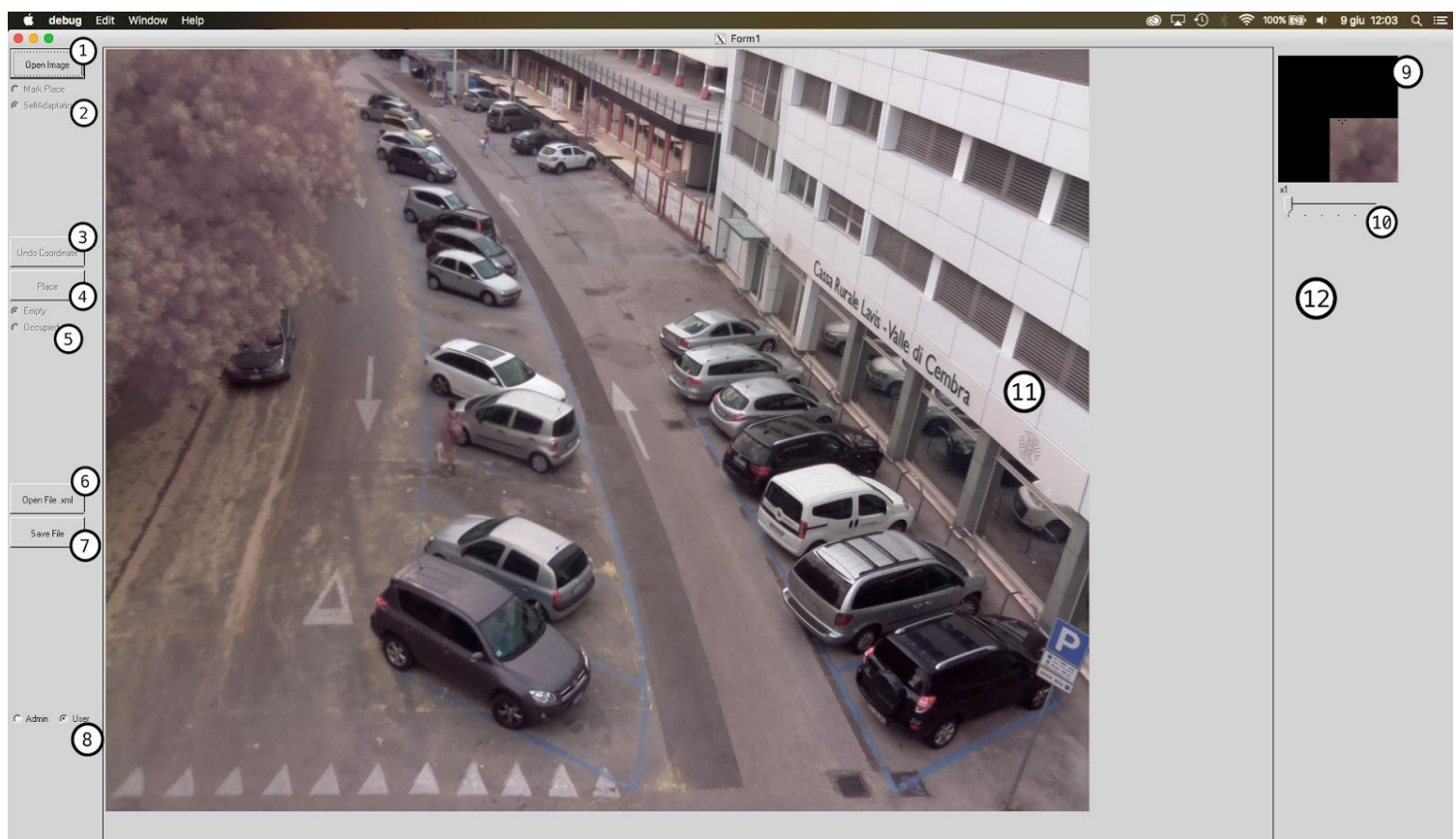
Descrizione Progetto:

Realizzare un'interfaccia per mappare i punti chiave degli stalli in un parcheggio.
Salvataggio dei punti chiave in un file XML e autoadattamento di questi ultimi in un'immagine elaborata (zoommata, traslata e ruotata).
Il programma è stato realizzato in C# utilizzando Visual C#.

Interfaccia:

L'interfaccia è divisa in amministratore e utente. Si presuppone che si utilizzi la parte destinata all'amministratore per impostare i 4 punti chiave di ogni stallo del parcheggio, per ognuno definire se è libero o occupato e per finire scegliere un punto da cui ricavare una buona porzione di immagine per l'autoadattamento. Infine si salvano tutti i dati in un file XML.

Utilizzando la parte utente è possibile aprire un'immagine, aprire un file XML creato dall'amministratore che eseguirà l'autoadattamento dei punti. Dopodiché l'utente potrà segnare se lo stato dello stallo è variato.



Modalità Admin:

- 1)Open Image: aprire l'immagine;
- 2)Mark Place: segnare i punti degli stalli;
SelfAdaptation Area: segnare l'area che sarà usata per l'autoadattamento;
- 3)Undo Coordinates: eliminare l'ultimo punto dello stallo segnato;
- 4)Place: marca uno stallo dopo aver segnato almeno 4 punti;
- 5)Empty: lo stallo marcato con Place è libero;
Occupied: lo stallo marcato con Place è occupato;
- 7)Save File: salva il file XML contenente le coordinate degli stalli e l'area di autoadattamento;
- 8) Passo in modalità Utente.

Modalità Utente:

- 1)Open Image: aprire l'immagine;
- 6)Open file .xml: apro il file xml contenente i dati degli stalli e dell'autoadattamento;
- 7)Save File: salva il file XML contenente le coordinate degli stalli e l'area di autoadattamento nel caso in cui l'utente abbia modificato lo stato degli stalli;
- 12)Dopo aver aperto il file, compaiono gli stati degli stalli ed è possibile cambiare gli stati degli stessi.

Altro:

- 11)Area in cui viene aperta l'immagine;
- 9)Area zoomata dell'immagine, in cui si possono centrare meglio i punti degli stalli;
- 10)Cursore per decidere quanto zoomare.

Spiegazione File XML:

Spiegazione riga per riga del file xml delle coordinate degli stalli:

- 1 - Autoadattamento: (X) (Y) → coordinate del punto scelto dall'admin per realizzare l'autoadattamento;
- 2 - Lotti: → inizio parte di interesse per l'utente;
- 3 - TotalePosti:(n) → numero di stalli totali;

- 5 - Posto:(id) → id dello stallo;
- 6 - Libero:(True/False) → indica se lo stallo è libero;
- 7 - Coordinata 0 X:(X) → coordinata x del primo punto dello stallo;
- 8 - Coordinata 0 Y:(Y) → coordinata Y del primo punto dello stallo;

- 10 - Coordinata 1 X:(X) → coordinata x del secondo punto dello stallo;
- 11 - Coordinata 1 Y:(Y) → coordinata Y del secondo punto dello stallo;

- 13 - Coordinata 2 X:(X) → coordinata x del terzo punto dello stallo;
- 14 - Coordinata 2 Y:(Y) → coordinata Y del terzo punto dello stallo;

- 16 - Coordinata 3 X:(X) → coordinata x del quarto punto dello stallo;
- 17 - Coordinata 3 Y:(Y) → coordinata Y del quarto punto dello stallo;

Dopo di che le righe dalla 5 alla 17 vengono ripetute un numero di volte pari al numero totale degli stalli.

```
1 Autoadattamento:942 171
2 Lotti:
3 TotalePosti:3
4
5 Posto:0
6 Libero:True
7 Coordinata 0 X:425
8 Coordinata 0 Y:436
9
10 Coordinata 1 X:643
11 Coordinata 1 Y:525
12
13 Coordinata 2 X:665
14 Coordinata 2 Y:603
15
16 Coordinata 3 X:425
17 Coordinata 3 Y:493
```

Autoadattamento:

Per realizzare l'autoadattamento abbiamo utilizzato wrapper multiplatforma chiamato Emgu CV; si tratta di un wrapper che permette di richiamare la libreria software Open CV nella suite Microsoft .Net rendendola compatibile con linguaggi di programmazione C#, VB etc.. Quindi è stato possibile compilare il wrapper usando l'IDE Visual C# per creare il programma sfruttando la potenza delle funzioni di computer vision di Open CV.

Primo approccio:

Come primo approccio avevamo pensato di utilizzare 3 punti dell'immagine cercando di prendere quelli che avessero più significato rispetto allo spazio colore sRGB. Una volta scelti i punti per ogni uno salvavamo i dati di altri 4 punti intorno a quello prescelto secondo una conformazione a croce.

Una volta salvati i dati utilizzavamo un algoritmo di ricerca a forza bruta, partendo dal punto originale e procedendo a spirale, per ritrovare i 3 punti sulla nuova foto. Ogni volta che l'algoritmo trovava un punto che possedeva gli stessi valori di quello originale confrontava anche gli altri 4.

Problemi:

Purtroppo ci siamo resi conto che l'utilizzo dei dati sRGB non è abbastanza robusto per identificare in modo univoco e certo tre punti, sarebbe servita una combinazione di più dati per rendere l'algoritmo più efficiente, ma questo avrebbe comportato un carico di elaborazione elevato e una gestione della memoria inefficiente.

Il secondo problema è che questo metodo è applicabile solamente alla stessa foto modificata attraverso un programma di computer grafica, perché presuppone che i colori rimangano identici tra una foto e l'altra cosa praticamente impossibile catturando due immagini in un parcheggio all'aperto.

Quindi abbiamo deciso di abbandonare questa strada.

Approccio definitivo:

Abbiamo optato per l'utilizzo della libreria software Open Cv in particolare l'utilizzo della classe SURF e dell'omonimo algoritmo SURF(Speeded Up Robust Features). Surf è l'evoluzione dell'algoritmo SIFT, viene utilizzato per identificare nelle immagini dei punti chiave e ottenerne le caratteristiche.

L'algoritmo SURF risulta essere di molto più veloce del SIFT e in media più robusto, dato che rispetto al predecessore approssima logaritmicamente i punti di interesse attraverso convoluzioni con filtri.

Descrizione algoritmo:

Per realizzare l'autoadattamento abbiamo deciso di acquisire una porzione della foto originale e salvarla. Quando l'utente sceglie la nuova immagine e carica il file XML creato al primo utilizzo, il bottone "Open File .xml", oltre che a caricare i dati richiama la funzione di autoadattamento che va a recuperare la porzione della foto salvata in precedenza e cerca i punti chiave che corrispondono tra le due immagini.

Una volta che l'algoritmo SURF ha trovato un numero sufficiente di punti con le stesse caratteristiche segnala sulla nuova immagine l'area corrispondente alla porzione di quella vecchia, una volta fatto questo riusciamo dall'area trovata a capire di quanto è traslata ruotata e zoommata la foto, potendo così applicare le corrette trasformazioni ai punti degli stalli.