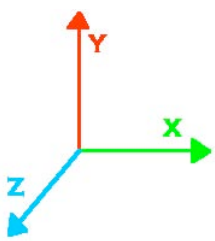
# PROGETTO DI ROBOTICA

Pranovi G. , Raimondi D.

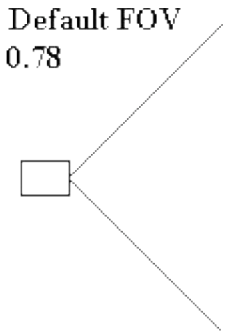
## 1. Simulazione di un mondo tridimensionale

La creazione del mondo tridimensionale da noi effettuata è stata realizzata in VRML 2.0 (Virtual Reality Model Language). La prima cosa necessaria è dunque un browser VRML che permetta di visualizzare il nostro mondo. La nostra scelta è caduta su un browser che opera come plugin di Interner Explorer o Netscape Navigator. Una volta che è stato installato e settato il browser VRML, l'utente può caricare i mondi VRML come fa con le pagine web e guardarli, spostandosi all'interno di essi come se vi camminasse, zoomando un particolare scelto , ruotando l'intero mondo e utilizzando altre opzioni che fanno parte dei tipi di navigazione disponibili selezionabili o con la Navigation toolbar o con la tendina dei comandi di navigazione che si apre cliccando sulla finestra col tasto destro del mouse. Per il corretto funzionamento del nostro programma è necessario nascondere la barra di navigazione. Per fare ciò dobbiamo caricare un qualunque file *.wrl, attivare la tendina dei comandi di navigazione, scegliere 'Options' e quindi il menù 'Worlds' dove si disattiva la casella 'Show Toolbar'.

Un mondo VRML è costituito da nodi, che sono tipi di oggetti. All'interno di questi nodi, vi sono dei campi, che sono proprietà dell'oggetto. I campi possono riguardare colore, forma, dimensione, traslazione e altre caratteristiche di un oggetto ma possono essere anche nodi figli all'interno del nodo principale a formare strutture geometriche composte. Il linguaggio fornisce le strutture geometriche comuni : spheres , cones , boxes e cilynders. Per disegnare strutture geometriche complesse viene utilizzato il nodo IndexFaceSet che crea le faccie della struttura collegando i vertici indicati. Le distanze in VRML sono misurate in metri. Tutti gli oggetti creati o importati da altri file .wrl con il nodo Inline, vengono posizionati nel centro del mondo. Il loro baricentro coincide con il baricentro della stanza. Da qui vengono scalati a proprio piacimento, ruotati e traslati nella posizione desiderata. Il sistema di coordinate VRML centrato nel baricentro del mondo opera come segue:




Il browser VRML crea un punto di entrata di default per la scena che è lungo l'asse Z positivo ad una distanza tale che l'intera scena sia visualizzata nella finestra. La posizione della camera può essere variata utilizzando il nodo Viewpoint . Questo nodo ha tre campi : position, orientation e fieldofView.

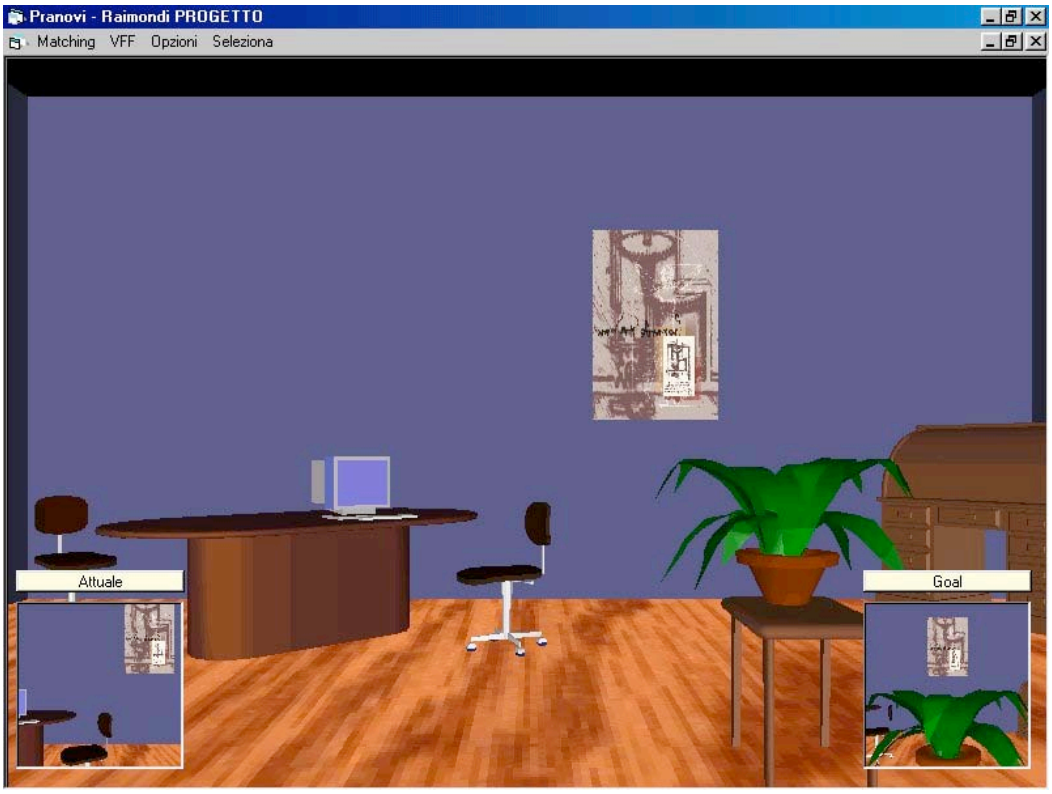



Il primo campo è una terna di valori floating point che rappresentano la posizione sull'asse X,Y e Z.
Il secondo campo è una terna di numeri floating point che definiscono la rotazione della telecamera.
Il terzo campo è un numero in radianti tra 0 e pi che rappresenta l'angolo di visuale.

Il browser visualizza il mondo nel punto di vista specificato che incontra per primo nel file, qualora vi siano più viewpoints.

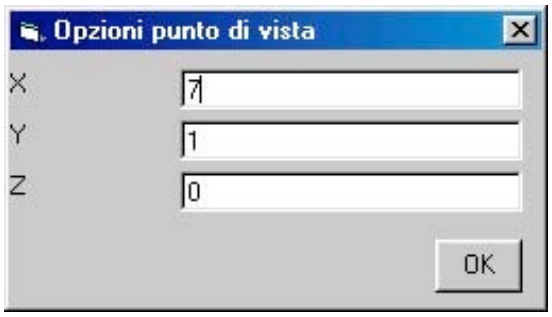
## 2. Interfaccia utente

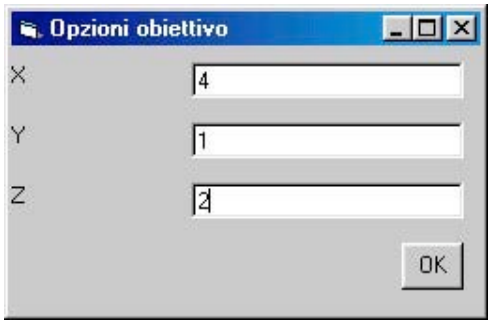
L'interfaccia utente è stata realizzata con Microsoft Visual Basic 6.0.
Ecco la schemata iniziale:




Cliccando con il mouse sullo schemo del computer posto sulla scrivania si può accedere alla relazione del programma in formato Html. Lanciando il file eseguibile 'robot' il programma carica immediatamente il mondo creato visualizzandolo dal punto di vista da noi stabilito (mondo.wrl). Inoltre crea due finestre, inizialmente vuote, di grandezza 128 x 128 pixels, una in basso a sinistra che conterrà la snapshot

visibile nel punto di partenza e una in basso a destra che conterrà la snapshot visibile nell'obiettivo. Spostando il puntatore sulle caselle "Attuale" o "Goal" si apre una tendina che permette di scegliere due opzioni : la prima consente di inserire un nuovo punto di partenza o un nuovo obiettivo; la seconda 'Prendi bmp' permette di leggere da video la snapshot memorizzando in una matrice 128 x128 i valori a 24 bit di ogni pixel. Anche dal menù a tendina 'Seleziona' si possono inserire visuale e obiettivo. Qui sotto sono riportati i form di inserimento di un nuovo punto visuale e obiettivo.




I numeri decimali devono essere inseriti con la virgola, mentre il campo Y, che rappresenta l'altezza è variabile ma deve essere uguale sia per il punto di vista che per l'obiettivo poiché abbiamo supposto che durante una navigazione la telecamera del robot non cambi la sua posizione verticale. L'altezza consigliata è di 1 m.
Il sistema di riferimento per l'inserimento di una nuova posizione è centrato nell'angolo in basso a sinistra con l'asse X positivo verso destra e l'asse Z entrante nello schermo. Una procedura provvede a convertire le coordinate dal sistema di riferimento dell'utente a quello del VRML. Il programma poi copia il file di base 'prova1.wrl'(default viewpoint) nel file 'provav.wrl' o 'provao.wrl', a seconda che si inserisca un nuovo punto di partenza o un nuovo obiettivo, lo apre e scrive in fondo ad esso il viewpoint scelto. Essendo quindi il primo e unico viewpoint che il browser legge nel file, visualizzerà il mondo da tale posizione con l'angolo visuale di default. Prima dell'inserimento di un nuovo punto di partenza o di un nuovo obiettivo, i files 'provav.wrl' o 'provao.wrl' vengono cancellati e si ripete il processo spiegato precedentemente. Le dimensioni della stanza da noi creata sono : 8 m di larghezza, 8 m di profondità e 4 m di altezza pertanto i punti che si possono selezionare devono stare all'interno di questo intervallo.

### 3. Il metodo del campo di forza virtuale

L'idea alla base di questo metodo è che gli ostacoli esercitino sul robot mobile una forza repulsiva mentre l'obiettivo produca una forza attrattiva. La prima condizione per utilizzare questo metodo è che sia nota la dimensione della stanza, la posizione degli oggetti all'interno di essa e l'obiettivo. Abbiamo quindi costruito una griglia

delle stesse dimensioni della stanza con una spaziatura di 0.5 m e possiamo visualizzarla come in figura :


Alla griglia corrisponde una matrice di occupazione 16 x 16 in cui a ogni cella occupata da un ostacolo è stato assegnato il valore –1. Sono state considerate ostacolo ovviamente anche le pareti per cui le righe e le colonne 0 e 15 sono state poste tutte a –1. Quindi ipotizzando di assimilare i punti della stanza alle celle in cui tali punti sono contenuti, abbiamo calcolato per ogni cella la forza risultante dovuta alla somma vettoriale delle forze repulsive e della forza attrattiva secondo le componenti X e Z.

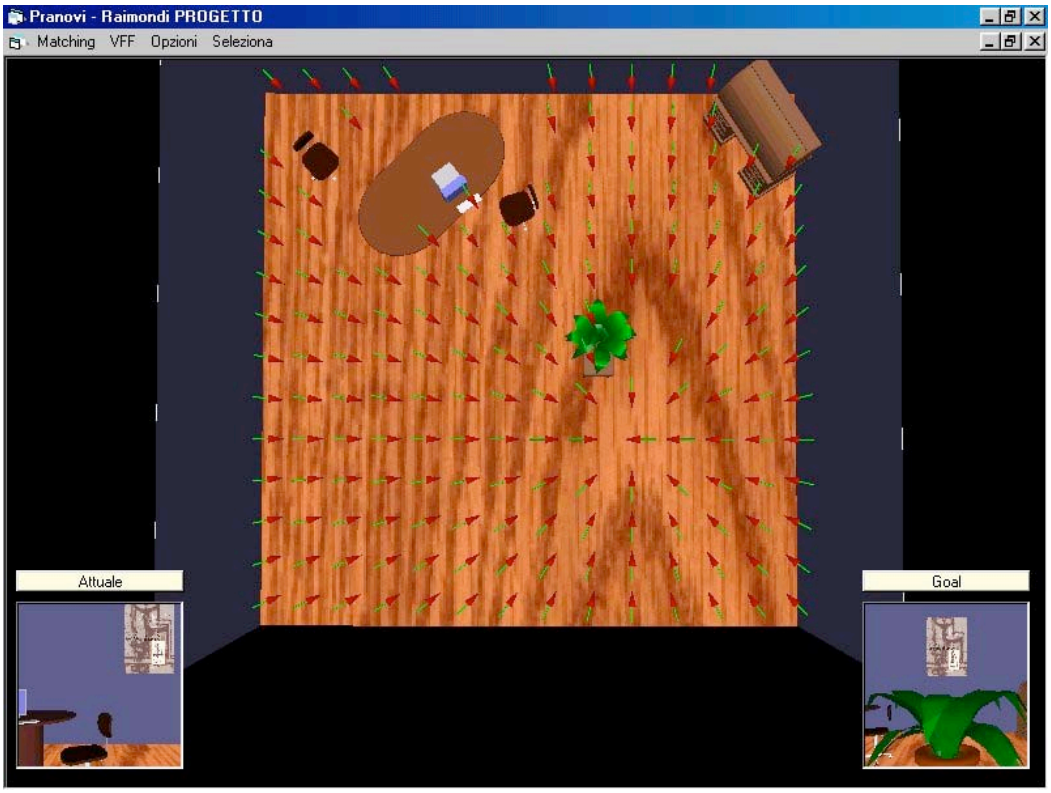
La forza attrattiva è la seguente : $$\boldsymbol{F}(i,j) = F_{ct}\left[\frac{x_t - x_0}{d(t)}\hat{x} + \frac{y_t - y_0}{d(t)}\hat{y}\right]$$

La forza repulsiva è data da : $$\boldsymbol{F}(i,j) = \frac{F_{cr}C(i,j)}{d^2(i,j)}\left[\frac{x_t - x_0}{d(i,j)}\hat{x} + \frac{y_t - y_0}{d(i,j)}\hat{y}\right]$$

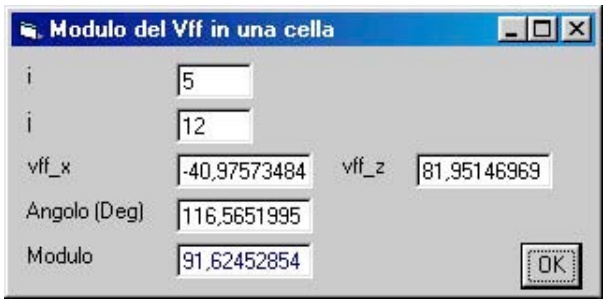
dove

$F_{ct}$ = costante di attrazione all'obiettivo
$F_{cr}$ = costante di repulsione
$x_0 y_0$ = coordinate attuali del robot
$x_i y_j$ = coordinate della cella(i , j)
$x_t y_t$ = coordinate dell'obiettivo
d(t) = distanza tra il robot e l'obiettivo
d(i , j) = distanza tra la cella(i , j) e il robot
C(i , j) = livello di occupazione della cella(i , j) (per noi C(i,j)=-1 sempre)

Note tali componenti è stato possibile inserire in una matrice per ogni cella l'angolo che la risultante forma con l'asse X visualizzando l'andamento dei versori come mostrato in figura :



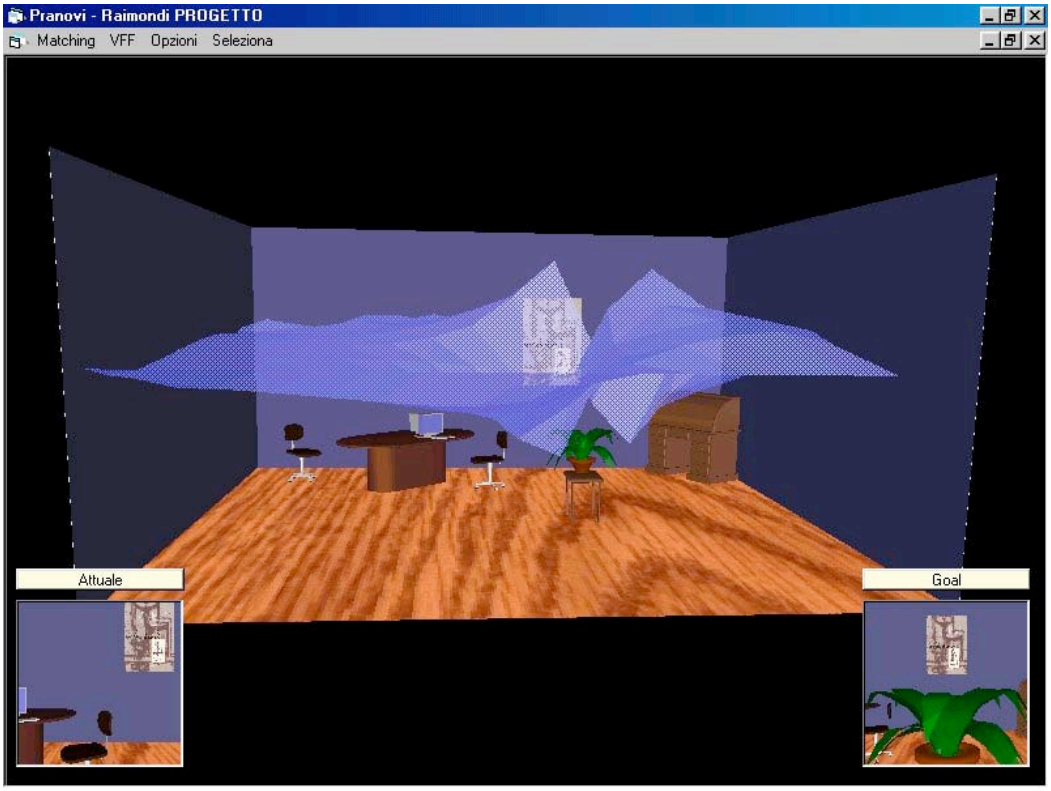

Ovviamente non è visualizzato il campo di forza nelle celle occupate dagli oggetti e nella cella obiettivo. Se si vuole conoscere il valore delle componenti del campo di forza virtuale, il modulo e la direzione del campo indicata dall'angolo che il versore forma con l'asse X in una singola cella, si può utilizzare l'opzione vff_cella del menù a tendina VFF e rappresentata qui sotto (con $0 < i < 15$ e $0 < j < 15$) :



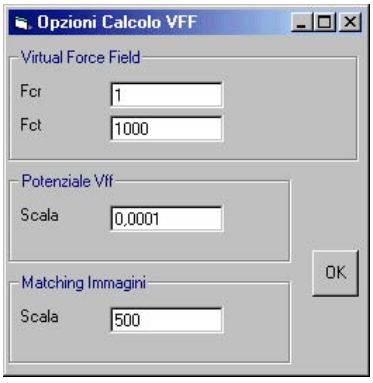

E' anche possibile visualizzare graficamente il modulo del campo di forza virtuale con l'opzione 'Modulo' del menù a tendina VFF.


Prima di effettuare il calcolo del potenziale, è necessario verificare la conservatività del campo. Se il campo è conservativo infatti esiste un'unica funzione potenziale data dall'integrale della forza virtuale lungo un percorso qualsiasi. Se il campo non è conservativo esistono infinite funzioni potenziali e il problema è complesso. Per determinare la conservatività abbiamo derivato in croce le componenti della VFF lungo l'asse X e l'asse Z rispettivamente rispetto a Z e a X applicando il rapporto incrementale. Le condizioni al contorno non possono essere valutate. La nostra matrice delle derivate avrà quindi una riga e una colonna in meno. Graficando con la funzione VRML ElevationGrid i risultati calcolati otteniamo una superficie di questo

tipo in cui non verrà visualizzato l'ultima colonna a partire da sinistra e l'ultima riga a partire dall'alto.




Se il campo è conservativo, noi possiamo scegliere il percorso più semplice costituito da un tratto orizzontale fino alla componente X dell'obiettivo e un tratto verticale fino alla componente Z dell'obiettivo o viceversa. L'integrale diventa così una sommatoria di alcune componenti della VFF lungo l'asse X e l'asse Z. Abbiamo visualizzato anche la superficie potenziale in base ai valori del modulo del potenziale memorizzati in una matrice. E' stato creato un form nel quale si possono cambiare i valori dei coefficienti di attrazione e repulsione e la scala della superficie lungo l'asse Z per qualunque grafico.



Con i valori di questi coefficienti scelti da noi si ottiene una superficie potenziale di questo tipo :


La funzione potenziale è tale che se lasciamo cadere una pallina in qualsiasi punto della superficie potenziale, questa rotola sempre verso la buca che si trova nel goal. (punto di equilibrio stabile)
E' possibile visualizzare anche contemporaneamente la superficie del modulo, quella di conservatività e la superficie potenziale. Infatti se dopo aver selezionato ogni opzione del menu VFF non applichiamo la procedura di Refresh che ci permette di ritornare al nostro mondo tridimensionale di partenza, tutte le opzioni successive che scegliamo vengono visualizzate insieme alla precedenti copiando in coda al medesimo file provag.wrl le nuove funzionalità da eseguire. Nell'immagine presentata sotto è visualizzata la superficie del modulo in rosso, la superficie di conservatività in blu e quella di potenziale in violetto ognuna con scala diversa per vederle tutte insieme all'interno della finestra. Era possibile aggiungere anche la griglia di suddivisione della stanza e le frecce del campo


## 4. Matching Snapshots

Un'altra tecnica per la navigazione di un robot mobile è implementata effettuando il confronto tra due immagini bidimensionali, in particolare tra la snapshot che si ottiene posizionandosi nel goal e la snapshot del punto di partenza. Questo metodo è del tutto indipendente dalla conoscenza delle dimensioni della stanza e dalle posizioni di partenza e obiettivo. La proiezione di oggetti tridimensionali in un immagine piana cambiando la posizione del punto di vista, genera delle distorsioni tra le varie immagini. Un modello che prende in considerazione le traslazioni,le compressioni e le rotazioni dovute a ciò è descritto dalle equazioni di Wu,Kittler. Tenendo conto però che gli oggetti del nostro mondo tridimensionale sono fissi e che il nostro robot non cambia la posizione verticale, si possono trascurare i parametri di rotazione e traslazione verticale, per cui il modello semplificato da noi utilizzato sarà il seguente:

$$\begin{cases} S_X(X,Y) = a_{0X} + a_{1X} \cdot X \\ S_Y(X,Y) = a_{2Y} \cdot Y \end{cases}$$

dove

(X,Y) sono le coordinate di un pixel della snapshot

$S_X(X,Y)\,S_Y(X,Y)$ sono le componenti del vettore spostamento in un immagine piana

$a_{0x}$ è il parametro di traslazione lungo l'asse X

$a_{1x}$ $a_{2y}$ sono i parametri di compressione lungo gli assi X e Y

Per ogni matching tra due immagini (attuale e goal) bisogna calcolare il valore dei tre parametri che minimizza l'Errore quadratico medio sull'intera immagine attuale.

**MSE** = $\frac{1}{M} \sum_{<x,y>\in S} E(X,Y)$

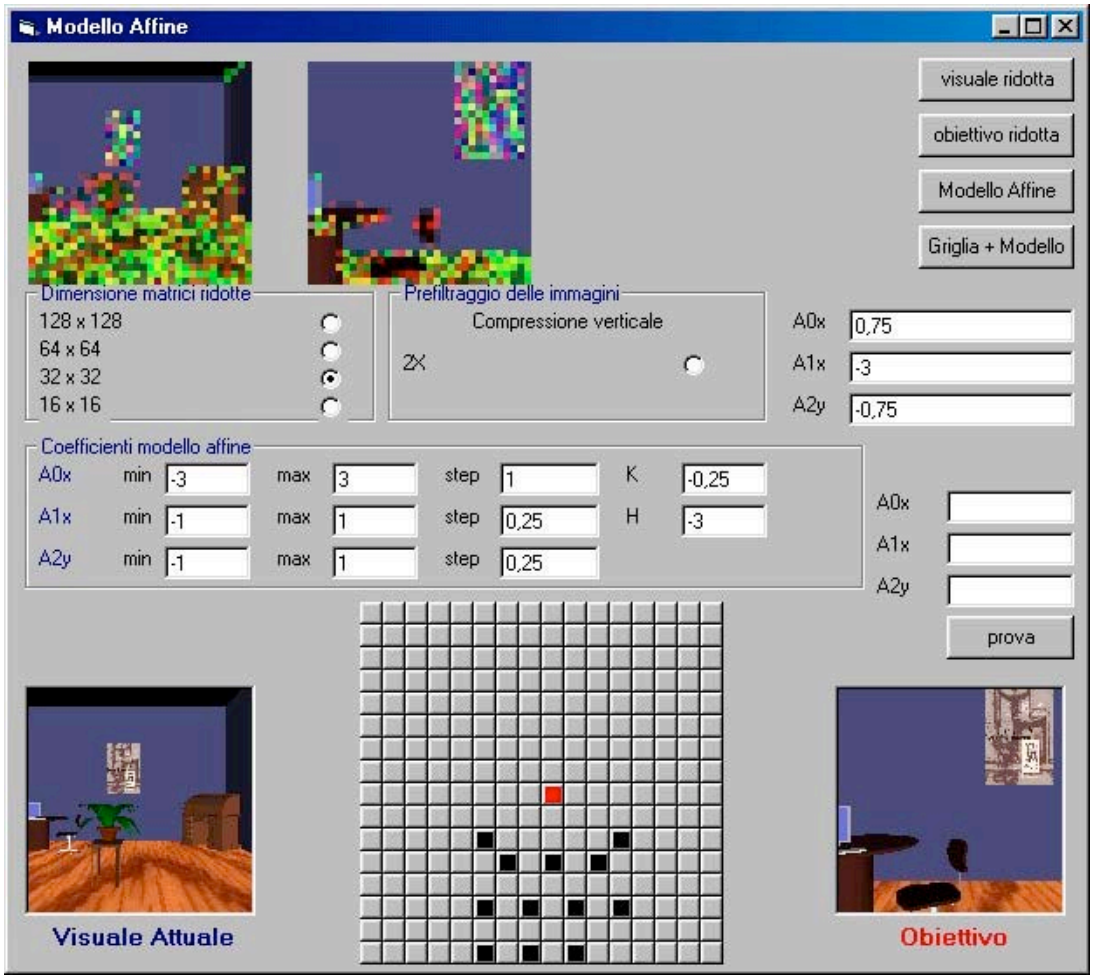
con E(X,Y) = $[I1(X,Y) - I2(X+S_x, Y+S_y)]^2$

dove

M è il numero di coppie $(X+S_x, Y+S_y)$

I1 è il valore del pixel (X,Y) della snapshot goal e I2 è il valore del pixel spostato della snapshot attuale

In sostanza il matching si propone di trovare una relazione tra gli oggetti o le porzioni degli oggetti comuni che compaiono sia nella snapshot attuale sia nel goal ma in posizioni e dimensioni differenti. Questa relazione è individuata dal vettore spostamento relativo a ciascun pixel dell'immagine attuale. L'obiettivo è quello di trovare il miglior vettore spostamento, cioè che sia considerato valido per l'intera immagine.
A livello di implementazione, il programma utilizza due API di Windows, GetDC per ottenere l'Handle delle finestre che contengono le immagini e GetPixel per memorizzare i valori a 24 bit dei pixel delle due immagini in due matrici 128X128.
Le due immagini possono essere memorizzate con risoluzioni diverse ( mediando i valori dei pixel adiacenti ) : 64x64 , 32x32 , 16x16. Questa opzione è stata inserita per velocizzare i calcoli del modello affine e verificare gli effetti che variazioni di risoluzione hanno sul modello stesso.
Il programma consente di variare gli intervalli dei parametri di traslazione e compressione, i coefficienti H e K e osservare la trasformazione dell'immagine attuale indotta dal modello affine inserendo direttamente i coefficienti da noi scelti.
Ma vediamo ora un esempio che ci permette di chiarire meglio quanto detto.




In questo esempio abbiamo scelto come obiettivo la posizione (4,1,4) che è rappresentato dalla cella rossa sulla griglia. I coefficienti del modello affine sono stati calcolati in tutte le posizioni contrassegnate dalle celle nere. Innanzitutto la posizione dell'obiettivo è scelta nella finestra principale del mondo e la snapshot obiettivo è riportata in questo form in basso a destra. Quindi dobbiamo premere il pulsante 'Griglia' che farà comparire una griglia 16x16 di pulsanti corrispondenti alla griglia di suddivisione della stanza e clicchiamo sui pulsanti desiderati per selezionare le celle da cui vogliamo prelevare le snapshots attuali. Il punto in cui si posiziona il viewpoint è il centro della cella 0,5x0,5m. Premiamo Ok per confermare la nostra scelta o chiudiamo il form per cancellare tutte le celle selezionate. A questo punto scegliamo la risoluzione su cui lavorare che potrà essere l'originale 128x128 oppure 64x64, 32x32 0 16x16. Se lo desideriamo possiamo applicare un prefiltraggio alle immagini, prima di eseguire il matching, che le comprime verticalmente di un fattore 2 ma è più conveniente lavorare sulle immagini reali. E' possibile cambiare i valori dei coefficienti del modello affine e il loro passo. Quelli che compaiono di defaullt sono stati già utilizzati da noi per effettuare varie prove.

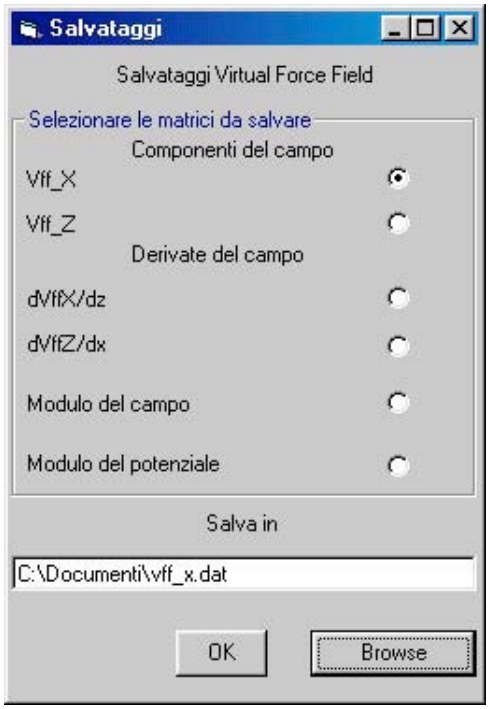
Quindi premendo il pulsante ‘modello affine’ si avvia il calcolo dei coefficienti del modello. Il programma carica la prima snapshot attuale nella finestra in basso a sinistra, legge a video i valori dei pixel che memorizza in una matrice quindi la trasforma nella risoluzione scelta e la mostra nella finestra visuale ridotta, poi inizia il matching con la snapshot obiettivo ridotta. A seconda della risoluzione scelta si dovrà attendere un tempo variabile per vedere visualizzato il risultato dell’elaborazione nelle caselle di testo dei parametri del modello affine. I valori delle componenti lungo l’asse X e Y del campo vettoriale di navigazione sono ottenuti moltiplicando i coefficienti H e K per i parametri A0x e A1x ottenuti.

$$\begin{cases} V_x = K \cdot a_{0X} \\ V_y = H \cdot a_{1X} \end{cases}$$

Tali valori vengono memorizzati in un’altra matrice, nell’elemento di indice corrispondente alla cella selezionata. Al termine del primo matching, il programma preleva la snapshot attuale dalla seconda cella selezionata e riprende lo stesso processo fino a completare tutte le celle scelte. L’utente vede quindi scorrere in sequenza l’immagine attuale, l’attuale ridotta e i valori dei parametri di ogni cella. Il versore del campo calcolato in ogni cella scelta viene poi visualizzato graficamente :

## 5. Salvataggi

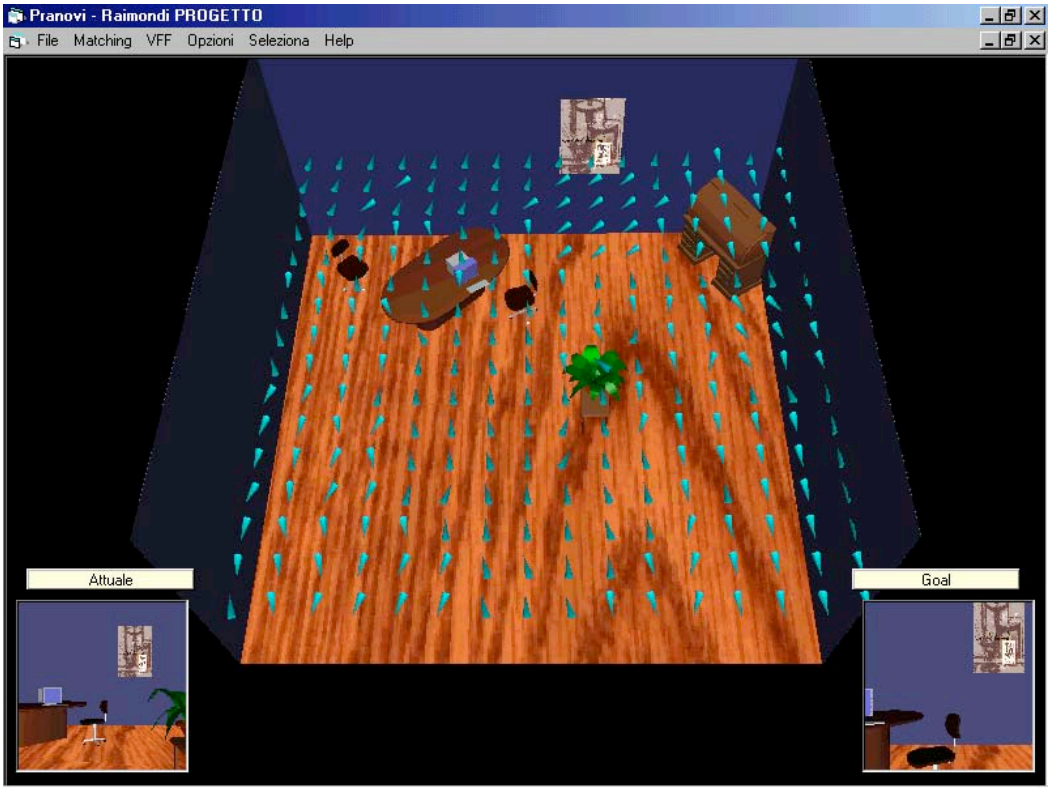
Per salvare in un file alcune delle matrici calcolate dal programma e che siamo interessati a visualizzare con il notepad o a importare in Matlab basta utilizzare l'opzione 'Salva'. Ecco la maschera:



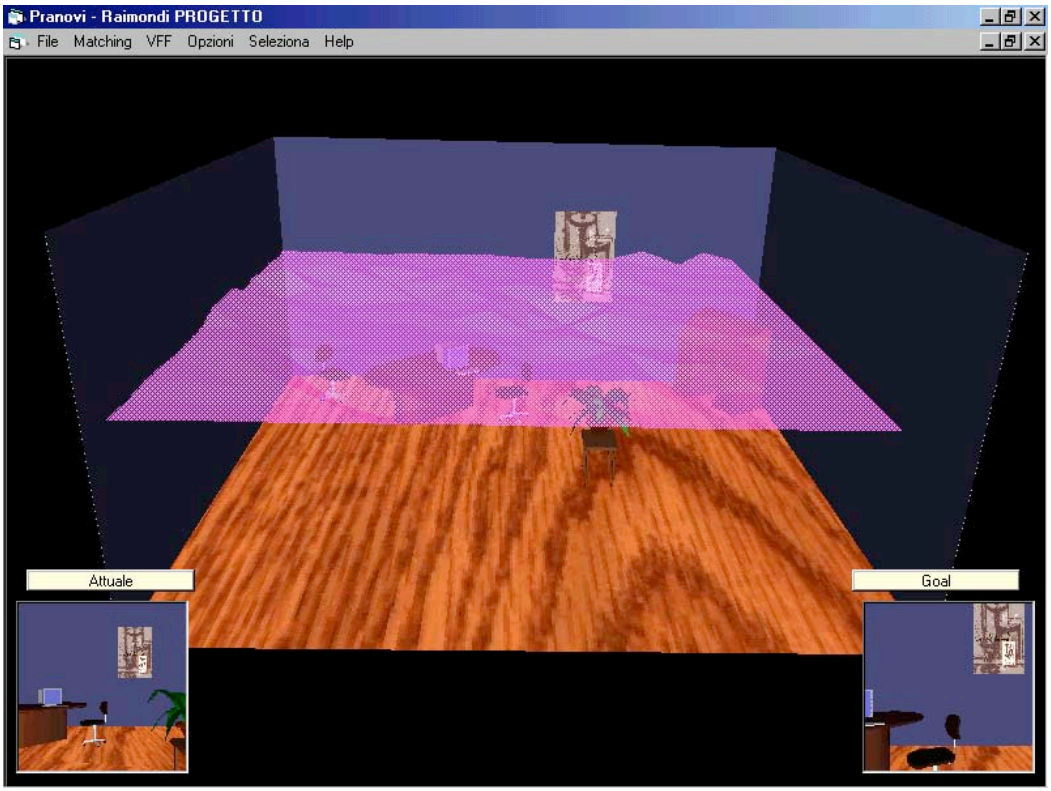

Selezionare una matrice alla volta, cliccare il pulsante Browse per scegliere la directory di destinazione e il nome del file con estensione a piacere (txt,dat,m,…).
Se vogliamo caricare la matrice in Matlab utilizziamo il comando load con path e nome del file. Matlab la memorizza in una variabile il cui nome è uguale al nome del file senza estensione.

## 6.**Esempio completo di matching**

Per testare il matching delle immagini è possibile caricare un esempio completo,dal menu "matching" selezionare "carica esempio".Vengono create le matrici contenenti le componenti del campo,calcolate le derivate parziali per la verifica della conservatività e valutato il potenziale risualtante utilizzando le stesse procedure spiegate per il Vff.
L'obiettivo è stato preso nel punto (4,1,4) con i coefficienti di default per il modello affine, eccetto per la risoluzione che è stata impostata a 32x32.
Il campo vettoriale risultante,considerando i confronti nelle 256 celle della griglia è il seguente:




Lasciamo all' occhio attento dell' osservatore ogni giudizio sulle direzioni dei versori Del campo e sulla conservatività dello stesso:




Nonostante il campo non appaia conservativo nell' intorno dell' obiettivo abbiamo ugualmente calcolato il potenziale con le semplificazioni spiegate in precedenza:


Pranovi - Raimondi PROGETTO
File Matching VFF Opzioni Seleziona Help
Attuale
Goal