Progetto MOSCARDO

Prototipo software per il riconoscimento di immagini e ricostruzione 3D (P3.1)

Confidenziale

Il Progetto MOSCARDO è realizzato con il determinante contributo della Regione Toscana a valere sul Programma Attuativo Regionale cofinanziato dal FAS (adesso FSC) e del contributo del MIUR a valere sui fondi FAR

Partner	*Autori (in ordine alfabetico)*
INFOMOBILITY S.r.l.	
EIS S.r.l.	
CNR-ISTI$_{WNLab}$	
CNR-ISTI$_{MMSLab}$	
CNR-ISTI$_{SILab}$	
DICEA UNIFI	

Sommario

1. Introduzione

L' obiettivo del presente documento è quello di fornire una descrizione del prototipo software per il riconoscimento e il monitoraggio a lungo termine della salute strutturale di alcuni siti storici: Fortezza Vecchia (Livorno) e Voltone (Livorno).

L'approccio scelto per la detection e l'analisi degli ammaloramenti presenti sulle strutture di interesse è quello image-based, che sfrutta l'installazione di marker nei punti individuati come critici o significativi ai fini del monitoraggio. Tale approccio, se da un lato offre la possibilità di una misurazione ripetibile e accurata, dall'altro pone una sfida tecnologica di rilievo sul numero di marker da usare rispetto all'estetica del bene culturale.

Inoltre, i dati raccolti (immagini, video) vengono utilizzati anche per la ricostruzione di un modello 3D dettagliato degli edifici storici in esame: la visualizzazione di acquisizioni effettuate in momenti temporali differenti consente un'immediata analisi preliminare di eventuali alterazioni riscontrate.

Nella **Sezione 2,** sono descritti i marker "ArUco" utilizzati. Vengono riportati, inoltre, esempi di punti critici nei quali tali markers verranno installati.

Nella **Sezione 3,** viene descritta brevemente la metodologia utilizzata, dalla calibrazione della camera utilizzata, all'acquisizione delle immagini e dei video mediante drone, all'elaborazione dei dati.

Nella **Sezione 4**, sono riportate le istruzioni d'uso del software, con un esempio di utilizzo.

2. Installazione marker

I marker che abbiamo scelto di utilizzare sono quelli presenti nella libreria software basata su openCV "Aruco" [1]. Come mostrato in Figura 1, sono marcatori planari quadrati bianchi e neri, facilmente rilevabili in modo robusto in un'ampia gamma di condizioni di luminosità.

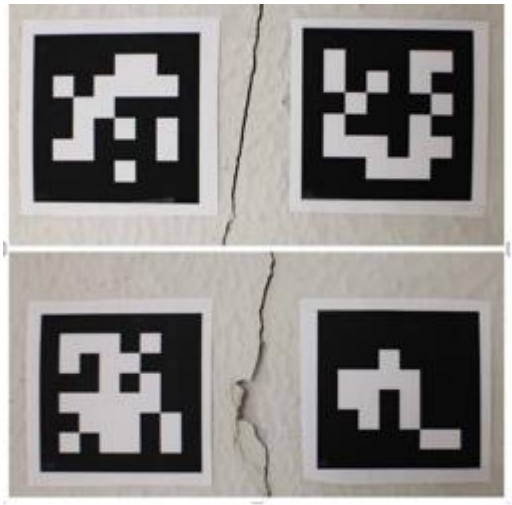


Figura 1- Due coppie di marker "ArUco" posizionati a cavallo di ammaloramemti superficiali

I marker, di dimensione 20x20 cm e stampati su pannelli di materiale plastico (PVC espanso o semi-espanso, Forex™), sono fissati in prossimità delle crepe da monitorare, in modo da essere utilizzati eventualmente anche come waypoint per il piano di volo degli UAV. L'idea è quella di installare alcune coppie di marker sui due lati della crepa di interesse, e di monitorare la variazione nel tempo della sua apertura come distanza tra due marker accoppiati. Tale approccio ripercorre quello presentato in **Error! Reference source not found.**.

In seguito a diversi sopralluoghi effettuati presso la Fortezza Vecchia e il Voltone a Livorno, ci sono state segnalate, da parte degli esperti, alcune crepe e difetti strutturali di notevole interesse.

2.2 La Fortezza Vecchia

Monumento simbolo della città di Livorno, è una struttura molto antica di grandissime dimensioni, a cielo aperto, e per questo soggetta a molteplici sollecitazioni atmosferiche e a degrado indotto sia dalle acque del mare, sia da cedimenti differenziali del terreno di fondazione. Inoltre, le sollecitazioni termiche provocano notevoli deformazioni stagionali.

Uno dei difetti strutturali di interesse è la crepa di grosse dimensioni e lunga diversi metri (mostrata in Figura 2) situata su una delle pareti a nord della Fortezza, all'incirca nel punto evidenziato sulla pianta in Figura 3Figura 3.

Figura 2- Difetto strutturale di grosse dimensioni presente su una delle pareti a nord della Fortezza Vecchia. Simulazione di posa marker ArUco

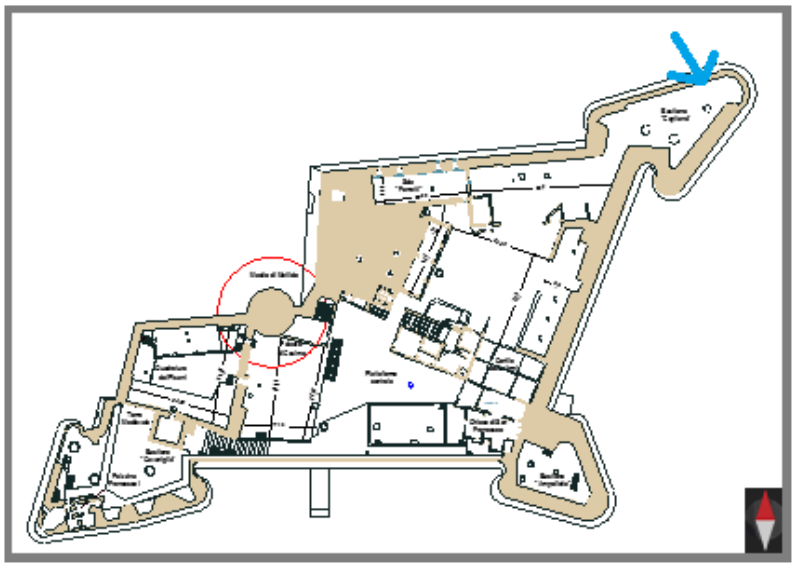


Figura 3 - Pianta della Fortezza Vecchia

2.3 Il Voltone

Il Voltone (**Error! Reference source not found.**) si estende sul Fosso Reale e sotto Piazza della Repubblica per circa 220m di lunghezza.

Figura 4- Il Voltone

Esso è soggetto sia a carichi permanenti, che a carichi variabili, ovvero quelli dovuti al traffico, alle estremità, e ai passanti, che in occasioni come concerti, eveti, manifestazioni, si concentrano principalmente nel centro della piazza.

Una delle crepe che intendiamo monitorare è raffigurata in Figura 5.

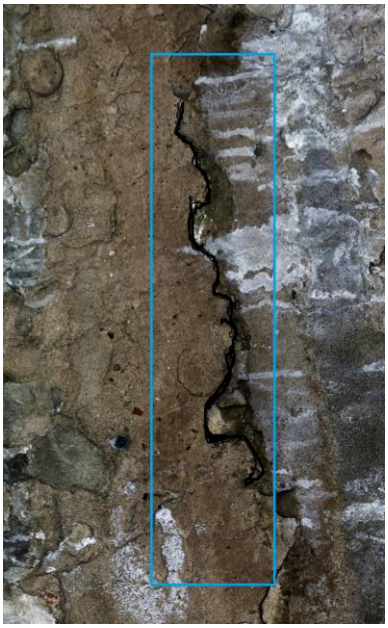


Figura 5- Uno degli ammaloramenti presenti sulla superficie del Voltone

3. Metodologia: cenni

1. Calibrazione della camera

La camera utilizzata per le acquisizioni è una Canon EOS M (18 Mega-pixel, mirror-less, con un sensore APS-C di 22.3x15 mm; massima risoluzione video 1920x1080 pixel a 30 fps; lunghezza focale variabile nel range 18-55 mm).

Al fine di determinarne i parametri interni di orientazione (lunghezza focale, le coordinate del punto principale, il coefficiente skew, ed i coefficienti di distorsione radiale e tangenziale) è innanzitutto necessaria una sua calibrazione, eseguita *una tantum*, per la rettifica dei dati acquisiti. I parametri estrinseci, ossia quelli associati alla posa della camera stessa, sono invece necessari alla ricostruzione della scena tramite fotogrammetria.

Per i test preliminari in laboratorio, sono state eseguite due calibrazioni della Canon Eos M: una prima, utilizzando la ChArUco *board* (Figura 6), e un'altra utilizzando la scacchiera standard. Nel primo caso, l'errore di ri-proiezione stimato è stato intorno a 1.55 px; nel secondo caso, migliore, pari a 0.76 px.

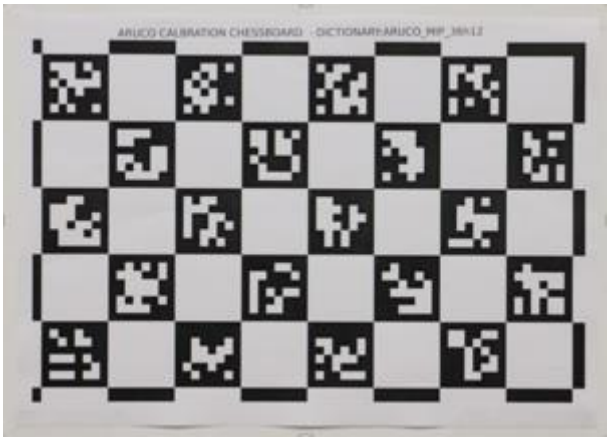


Figura 6- Scacchiera ChArUco

2. Acquisizione immagini e video

Il sistema di acquisizione dati è montato su un drone, un esacottero progettato e assemblato da ISTI-CNR. Tra le due modalità di volo, standard e programmata, verrà utilizzata principalmente la seconda, dal momento che permette di effettuare la stessa rotta di volo in spazi temporali diversi, ottenendo acquisizioni molto più confrontabili. La distanza tra il drone e la superficie di interesse sarà circa 2-2.5m. In base ai risultati derivanti dai test preliminari eseguiti in laboratorio (ambiente controllato), il numero di frames acquisiti sarà maggiore o uguale a 6.

I dati acquisiti saranno trasmessi all' unità di ricezione a terra (laboratorio mobile su autovettura attrezzata) mediante connessione ibrida wireless/4G in modo da renderli disponibili on-line da qualunque sito geografico. Algoritmi di SLAM saranno infine utilizzati per la localizzazione real time del drone (e dunque visualizzazione) sulla consolle dell'operatore a terra.

3. Elaborazione dei dati

***Simultaneous Localization and Mapping* (SLAM) basato su markers**

L'approccio seguito riprende ed estende quello descritto da Munoz et al. [3].

Tale approccio prevede:

- acquisizione di un set di immagini (n. frames maggiore o uguale a 6);

- acquisizione dei parametri di calibrazione della camera;
- detection dei marker mediante libreria ArUco;
- *3D pose estimation* dei markers per ciascun frame e ottimizzazione dei parametri della matrice di trasformazione dal sistema di riferimento dei markers al sistema di riferimento globale, ovvero minimizzazione dell'errore di riproiezione nel calcolo delle coordinate 3D dei vertici dei markers;
- calcolo del baricentro di caiascun marker;
- calcolo della matrice delle distanze tra ciascuna coppia di marker.

Ricostruzione 3D degli edifici storici di interesse

Per la ricostruzione 3D dei siti storici di interesse viene utilizzato Agisoft Photoscan (www.agisoft.com), uno dei più accurati software di fotogrammetria. Esso ricostruisce la scena in diversi passi consecutivi: i) estrazione dei punti caratteristici; ii) correlazione di questi in tutte le immagini in modo da ottenere i punti di controllo (le corrispondenze tra le varie immagini); iii) da questa prima ricostruzione sparsa, è calcolata una nuvola densa di punti; iv) da questo output intermedio si arriva alla superficie poligonale (mesh), ovvero il modello 3D. Un esempio di ricostruzione 3D è riportato in Figura 7.

Inoltre, mediante il motore grafico Unity (www.unity3d.com), viene creata una scena virtuale contente la struttura storica ricostruita, al fine di facilitare e rendere *engaging* la "visita" del sito storico. All'interno della scena, infatti, l'utente può facilmente navigare intorno all'oggetto ricostruito ed effettuare una rapida analisi di tutte le aree di interesse della struttura. Nell'ambiente virtuale, i difetti strutturali, le crepe e gli ammaloramenti sono evidenziati ed etichettati con le ultime misurazioni. È anche possibile recuperare le analisi precedenti o visualizzare grafici che rappresentano l'evoluzione dell'apertura delle crepe nel tempo.

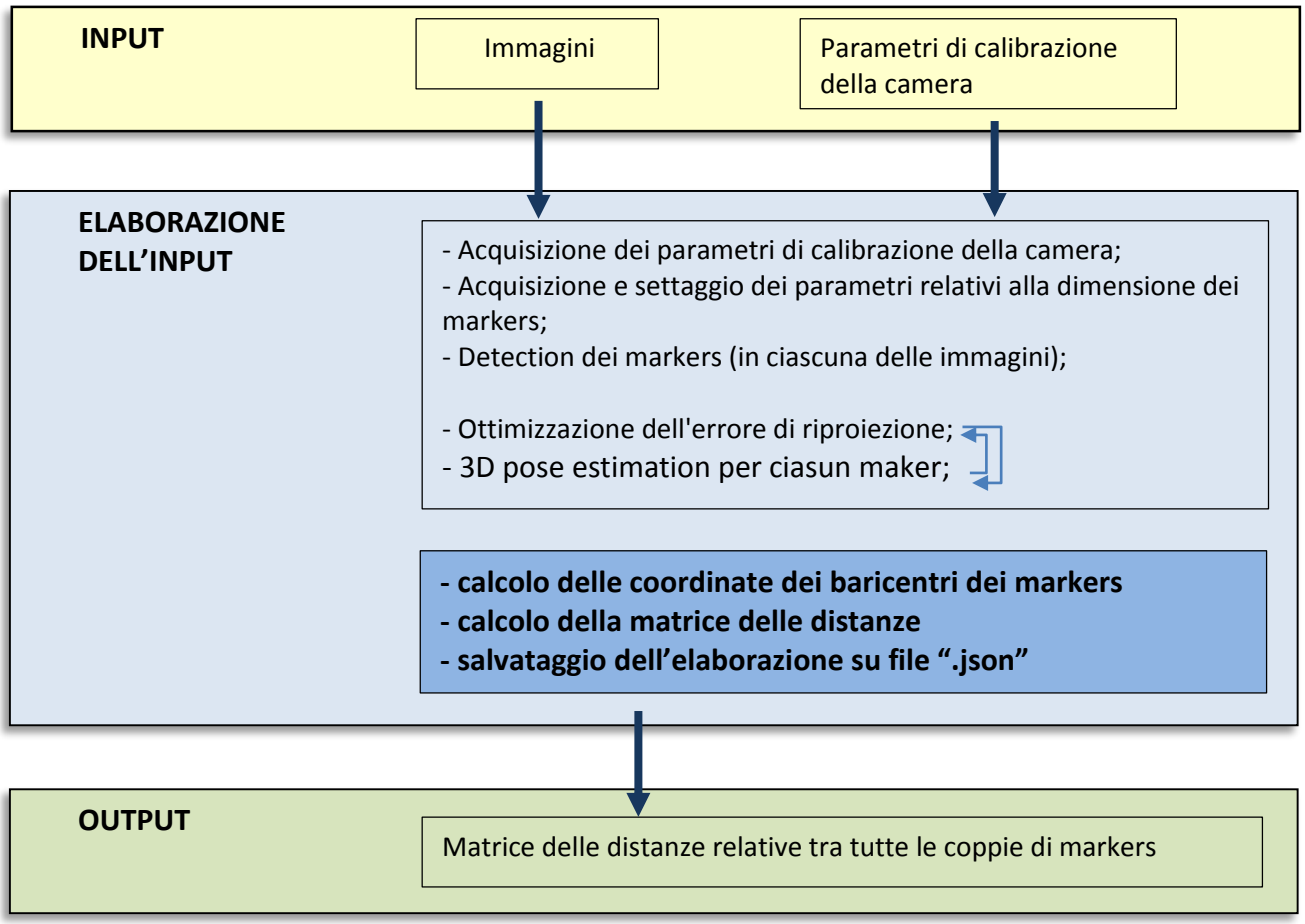
Figura 7- Ricostruzione 3D di una torretta

3.1 Requisiti e Funzionalità

3.2 Codice (pseudo codice) [QUI O IN APPENDICE]

Riportiamo qui il *work-flow* del programma.

Il nostro software riprende ed estende quello presentato in [3]. Il codice aggiuntivo, da noi implementato, è riportato in appendice.



4. Istruzioni d'uso [CON ESEMPIO: INPUT, STRINGA COMANDO, OUTPUT]

Vengono qui riportate le istruzioni per il corretto utilizzo dell'algoritmo precedentemente descritto. Per maggiore chiarezza, viene riportato un esempio relativo ai test sperimentali effettuati in ambiente controllato.

1. Input

```
Usage: directoryWithImages cameraparameters.yml marker_size(meters) dictionary out [-ref id]
```

In ingresso, il programma ci chiede di inserire:

- il path della directory contenente il set di immagini da elaborare;
- file con estensione ".yml" di calibrazione della camera;
- la dimensione del lato dei markers, in metri;
- il dizionario di markers utilizzato, tra quelli presenti (ARUCO, ARUCO_MIP_16h3, ARUCO_MIP_25h7, ARUCO_MIP_36h12, ARTOOLKITPLUS, ARTOOLKITPLUSBCH, TAG16h5, TAG25h7, TAG25h9, TAG36h11, TAG36h10, CHILITAGS, ALL_DICTS);
- il nome del file di output;
- l'identificativo dei markers presenti nelle immagini e che dovranno essere riconosciuti (input opzionale)

Nel nostro caso, le immagini (che riportiamo, come esempio, in Figura 8) erano contenute nella directory "home/ubuntu/MOSCARDO/Esperimenti/coord/T1".

Figura 8- Simulazione di apertura crepa in laboratorio (set di 6 immagini)

Il lato di ciascuno dei marker era di 0.55m (5.5cm); il dizionario utlizzato era "ARUCO_MIP_36h12".
In appendice è riportato il file di calibrazione della camera con estensione ".yml" .

2. Stringa di comando

Dunque, la stringa di comando in ingresso al programma:

```
home/ubuntu/MOSCARDO/Esperimenti/coord/T1 StandChessbCalib.yml 0.055 ARUCO_MIP_36h12 T1_distance
```

3. Output

In output avremo il file con estensione ".json", contenente le coordinate 3D dei singoli markers, le coordinate 3D del baricentro di ciascun marker e le distanze relative.


```
{
    "matrix aruco_bc_nmarkers": 6,
    "matrix aruco_bc_mInfoType": 1,
    "matrix aruco_bc_cornersEbaricentri": [
        { "id": 0, "corners": [ [ -2.7499999850988388e-02,
                    2.7499999850988388e-02, 0.0 ], [
```

```
                    2.7499999850988388e-02, 2.7499999850988388e-02, 0.0
],
                [ 2.7499999850988388e-02, -2.7499999850988388e-02, 0.0 ],
                [ -2.7499999850988388e-02, -2.7499999850988388e-02, 0.0 ]
],
            "baricentro": [ 0.0, 0.0, 0.0 ] },
        { "id": 1, "corners": [ [ -4.1390985250473022e-01,
                    -8.3671078085899353e-02, -1.4874526299536228e-02 ],
                [ -3.5891419649124146e-01, -8.3655908703804016e-02,
                    -1.4185366220772266e-02 ], [ -3.5890746116638184e-01,
                    -1.3865178823471069e-01, -1.3512532226741314e-02 ],
                [ -4.1390311717987061e-01, -1.3866695761680603e-01,
                    -1.4201692305505276e-02 ] ], "baricentro": [
                -3.8640865683555603e-01, -1.1116143316030502e-01,
                -1.4193529263138771e-02 ] },
        { "id": 2, "corners": [ [ -2.7703931555151939e-02,
                    -8.2772336900234222e-02, -1.4143157750368118e-03 ],
                [ 2.7283286675810814e-02, -8.2301765680313110e-02,
                    -3.2600702252238989e-04 ], [ 2.7759244665503502e-02,
                    -1.3729906082153320e-01, -5.9373769909143448e-04 ],
                [ -2.7227973565459251e-02, -1.3776962459087372e-01,
                    -1.6820464516058564e-03 ] ], "baricentro": [
                2.7656555175781250e-05, -1.1003570258617401e-01,
                -1.0040267370641232e-03 ] },
        { "id": 4, "corners": [ [ -4.1345345973968506e-01,
                    8.1003747880458832e-02, -1.7969015985727310e-02 ], [
                    -3.5854038596153259e-01, 8.0810576677322388e-02,
                    -1.4883665367960930e-02 ], [ -3.5882306098937988e-01,
                    2.5834267958998680e-02, -1.3294595293700695e-02 ], [
                    -4.1373613476753235e-01, 2.6027439162135124e-02,
                    -1.6379946842789650e-02 ] ], "baricentro": [
                -3.8613826036453247e-01, 5.3419008851051331e-02,
                -1.5631806105375290e-02 ] },
        { "id": 6, "corners": [ [ -1.3719843327999115e-01,
                    8.1125393509864807e-02, -2.6271417737007141e-03 ], [
                    -8.2203567028045654e-02, 8.1293657422065735e-02,
                    -1.8947350326925516e-03 ], [ -8.2031860947608948e-02,
                    2.6294523850083351e-02, -2.1517379209399223e-03 ], [
                    -1.3702672719955444e-01, 2.6126259937882423e-02,
                    -2.8841446619480848e-03 ] ], "baricentro": [
                -1.0961514711380005e-01, 5.3709961473941803e-02,
                -2.3894398473203182e-03 ] },
        { "id": 7, "corners": [ [ -3.0400955677032471e-01,
                    2.5502916425466537e-02, -1.2234330177307129e-02 ], [
                    -2.4906121194362640e-01, 2.5398228317499161e-02,
                    -9.8535232245922089e-03 ], [ -2.4919222295284271e-01,
                    -2.9598284512758255e-02, -9.2480480670928955e-03 ],
                [ -3.0414056777954102e-01, -2.9493596404790878e-02,
                    -1.1628855019807816e-02 ] ], "baricentro": [
                -2.7660089731216431e-01, -2.0476840436458588e-03,
                -1.0741189122200012e-02 ] }
    ],
    "matrix aruco_bc_BaricentriDistances": [
        { "id1": 0, "id2": 1, "distance": 4.0233067316904270e-01 },
        { "id1": 0, "id2": 2, "distance": 1.1004028661452461e-01 },
        { "id1": 0, "id2": 4, "distance": 3.9012908118479478e-01 },
```

```
        { "id1": 0, "id2": 6, "distance": 1.2208992530547433e-01 },
        { "id1": 0, "id2": 7, "distance": 2.7681694772465781e-01 },
        { "id1": 1, "id2": 2, "distance": 3.8666297280829343e-01 },
        { "id1": 1, "id2": 4, "distance": 1.6458694114279007e-01 },
        { "id1": 1, "id2": 6, "distance": 3.2239193586172810e-01 },
        { "id1": 1, "id2": 7, "distance": 1.5484015552578664e-01 },
        { "id1": 2, "id2": 4, "distance": 4.1958971626708208e-01 },
        { "id1": 2, "id2": 6, "distance": 1.9706879180724823e-01 },
        { "id1": 2, "id2": 7, "distance": 2.9711879859939849e-01 },
        { "id1": 4, "id2": 6, "distance": 2.7684016519841498e-01 },
        { "id1": 4, "id2": 7, "distance": 1.2287760601353317e-01 },
        { "id1": 6, "id2": 7, "distance": 1.7624672340542322e-01 }
    ]
}
```

5. Appendice

1. Esempio di file di dati di calibrazione della camera

```
%YAML:1.0
---
calibration_time: "Thursday 14 June 2018 08:26:31 PM IST"
image_width: 5184
image_height: 3456
flags: 0
camera_matrix: !!opencv-matrix
   rows: 3
   cols: 3
   dt: d
   data: [ 5.44387365e+03, 0., 2.65831310e+03, 0.,
       5.46243673e+03, 1.66002936e+03, 0., 0., 1. ]
distortion_coefficients: !!opencv-matrix
   rows: 1
   cols: 5
   dt: d
   data: [ -1.1595e-01, 1.2589e-01,
       5.9e-04, 1.35e-03,
       0. ]
   avg_reprojection_error: 7.621756e-01
```

2. Codice implementato

```
void MarkerMap::saveDistanceMatrixToFile(string sfile)
    {
        cv::FileStorage fs(sfile, cv::FileStorage::WRITE);
        saveDistanceMatrixToFile(fs);
    }
    /**Saves the board info to a file
    */
    void MarkerMap::saveDistanceMatrixToFile(cv::FileStorage& fs)
    {

        fs << "matrix aruco_bc_nmarkers" << (int)size();
        fs << "matrix aruco_bc_mInfoType" << (int)mInfoType;


      std::vector<cv::Point3f> baricentri;
      cv::Point3f temp;

      //calcola baricentri
       for (size_t i = 0; i < size(); i++)
       {
             temp.x = 0.0;
             temp.y = 0.0;
             temp.z = 0.0;

             for (size_t c = 0; c < at(i).size(); c++)
             {
                   temp.x += at(i)[c].x;
                   temp.y += at(i)[c].y;
                   temp.z += at(i)[c].z;
```

```
            }
            temp.x /= at(i).size();
            temp.y /= at(i).size();
            temp.z /= at(i).size();

            baricentri.push_back(temp);
        }


//        fs << "matrix aruco_bc_dict" << dictionary;

        fs << "matrix aruco_bc_cornersEbaricentri"
           << "[";
        for (size_t i = 0; i < size(); i++)
        {
            fs << "{:"
               << "id" << at(i).id;

            fs << "corners"
               << "[:";
            for (size_t c = 0; c < at(i).size(); c++)
                fs << at(i)[c];
            fs << "]";
            fs << "baricentro" << baricentri.at(i);
            fs << "}";
        }
        fs << "]";

        fs << "matrix aruco_bc_BaricentriDistances"
           << "[";
        for (size_t i = 0; i < size(); i++)
        {
        for (size_t j = i+1; j < size(); j++)
        {

            fs << "{:"
               << "id1" << at(i).id;
              fs << "id2" << at(j).id;

              temp = baricentri.at(i)-baricentri.at(j);

            fs << "distance" << cv::norm(temp);
//            fs << "distance" << cv::norm(baricentri.at(i));

            fs << "}";
       }
        }
        fs << "]";



       }
```

Riferimenti bibliografici

[1] S. Garrido-Jurado, R. Munoz-Salinas, F.J. Madrid-Cuevas, M.J. Marin-Jimenez, “Automatic generation and detection of highly reliable fiducial markers under occlusion”, Pattern Recognition, vol. 47 (6), 2014, pp. 2280-2292

[2] S.Nishiyama, N. Mianakata, T. Kikuchi, T.Yano, Improved digital photogrammetry technique for crack monitoring, Advanced Engineering Informatics, 2015, 29, 851-858

[3] Munoz-Salinas, R.; Marin-Jimenez, M.J. andYeguas-Bolivar, E.; Medina-Carnicer, R. Mapping and localization from planar markers. P. Recog. 2018, 73, 158–171.