Università degli Studi di Firenze

DIPARTIMENTO DI MECCANICA E TECNOLOGIE INDUSTRIALI

DOTTORATO DI RICERCA IN PROGETTO E SVILUPPO DI PRODOTTI E PROCESSI INDUSTRIALI

SETTORE SCIENTIFICO DISCIPLINARE ING-IND/15

CICLO XXV

TECNICHE E METODI PER L'AUTOMAZIONE DEL PROCESSO DI TRADUZIONE DEI DISEGNI TECNICI DI STILE IN GEOMETRIE TRIDIMENSIONALI DI TIPO "FREE-FORM"

DOTTORANDO:

ING. MATTEO PALAI

TUTOR:

PROF. ING. LAPO GOVERNI

CONTRORELATORE:

PROF. ING. LUCA DI ANGELO

COORDINATORE DEL DOTTORATO:

PROF. ING. MARCO PIERINI

ANNI 2010/2012

A Federica,
perché tutto questo
è anche merito tuo.

A questo punto credo proprio di aver finito.

La mia "carriera" studentesca, almeno quella ufficiale, quella delle maestre, dei professori, dei "compagni di classe" è terminata. A tutti loro che fin da quando ero bambino e andavo a scuola a Cireglio con la cartella rossa mi hanno aiutato ad arrivare fin qui voglio dire grazie.

Un ringraziamento speciale va alla mia famiglia, sempre attenta in tutti questi anni a non farmi mai mancare nulla, a fare in modo che potessi fare tutte le scelte che di volta in volta ritenevo migliori senza mai impormi nulla ma aiutandomi sempre e in ogni modo. Grazie Babbo, grazie Mamma, grazie Luca e grazie ai miei sei Nonni, Ardito e Foscarina, Amos e Bice (Beatrice), Romano (Lorenzo) e Romana.

Infine, last but not least, voglio ringraziare le persone con cui ho trascorso gli ultimi, intensi, anni di Università, quelle con cui ho vissuto tanti bei momenti che sicuramente rimarranno nei miei ricordi, grazie a Lapo, a Yary, a Rocco, a Monica, a Maurizio, a Matteo, a Iacopo, a Mirco e a Luca.

A tutti voi,
grazie.

Matteo

Abstract

Computer Aided Design systems are deemed to be essential for all the phases characterizing the design and the development of a new industrial product. However, especially for products characterized by a strong stylistic content, handmade drawings are often preferred to CAD software packages.

In the early stage of the development of a new product, designers typically produce a rich set of sketches (usually in the form of orthographic projections) to develop and communicate their ideas. Product-managers often have to select stylistic alternatives based on a set of hand-drawings depicting possible solutions. However, it is much more effective to base the selection on a virtual 3D model which conveys far more information. In fact some hand-drawn alternatives are even "translated" into 3D models (and possibly rendered models) capable to provide a more realistic view of the object and to allow a deeper analysis of the stylistic design.

The translation process, involving a close interaction of stylistic designer and CAD operators in order to produce a CAD model carefully representing the designer's intent, is known to be considerably time consuming. Accordingly, the design alternatives available in the form of three-dimensional models result to be a very small subset of those developed by the designers (usually one or, at most, two).

The common methodology (see Figure A1) used to turn a set of orthographic projections into a 3D model starts from arranging the scanned images of the hand-drawn views on the correspondent orthogonal planes in a CAD software environment. Using such images, the CAD operator (often assisted by the designer) manually redraws the style lines in order to obtain the 3D wireframe which is eventually used as a support frame for the definition of the final surfaces.

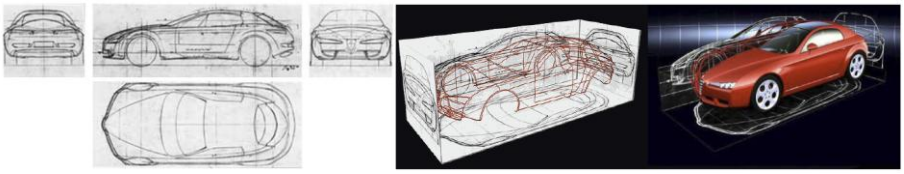


Figure A1: Typical 2D to 3D "translation" process (courtesy of Italdesign-Giugiaro).

The automation of this process, i.e. the 3D retrieval from 2D drawings (known as “reconstruction problem”), is a key target for commercial software houses as well as a vigorous focus from an academic outlook.

In order to cope with this issue a number of works have been proposed since first 1970s, providing a series of methodologies for solving the reconstruction problem.

Recently a set of software tools have also been released by major software houses, like Dassault Systemes®, Autodesk® and PTC®, which support the CAD operators in some of the reconstruction process phases. These tools, however, entail a strong user interaction and only marginally speed up the process. In addition, most of them require as an input an “exact” set of 2D vector drawings (e.g. DXF files).

Moving from these considerations, this work is meant to discuss a methodology to perform a quasi-automatic "translation" starting from a 2D scanned image of hand-drawn orthographic projections.

The proposed methodology to perform the 3D model reconstruction is structured according to a set of phases which are briefly described in this section.

BUILDING A WIREFRAME-LIKE VOXEL CLOUD (3D IMAGE) REPRESENTING A RASTER MODEL OF THE ORIGINAL OBJECT

1) The Image of the hand-drawn sheet depicting the orthographic projections is acquired by means of a flatbed scanner (grayscale – 8bit).

2) The orthographic views in the resulting image are identified and the sheet orientation is automatically compensated. In this phase, it is assumed that the views are correctly arranged in the original drawing (according either to the first-angle or to the third-angle projection rule) and separated by means of continuous lines representing the mutual intersections of the planes on which the views are projected. In this phase the object’s bounding box dimensions (in pixel) are also computed.

3) Each single view is considered as a separate image and the outlines undergo a grayscale dilation using a kernel whose dimension is set proportionally to the expected error in correspondence among the projections (for further explanation see also phase 5). After dilation, the images are inverted and normalized.

4) The images are extruded orthogonally to the image plane to cover the entire bounding box computed in phase 2. A set of 3D images is, thereby, obtained with each voxel having a real value between 0 and 1.

5) The entire set of 3D images is multiplied element by element so to obtain a final 3D image G, where the object outlines are represented by a 3D

wireframe-like voxel cloud (see Figure A2). A given region of the 3D outline is correctly reproduced in G if sufficiently good matching is found between the corresponding 2D outlines in the original views (perfect match: G(i,j,k) = 1; partial match: 0 < G(i,j,k) < 1; no match: G(i,j,k) = 0). The amount of acceptable error is given by the dilation kernel dimension. In this phase, some spur voxel branches may be generated in the 3D model, due to falsely matching 2D profiles. In fact, since finite precision is used, the extrusion of some profiles may practically overlap in the 3D space even if, ideally, they should not. These artifacts need to be manually removed prior to continue with the rest of the procedure.

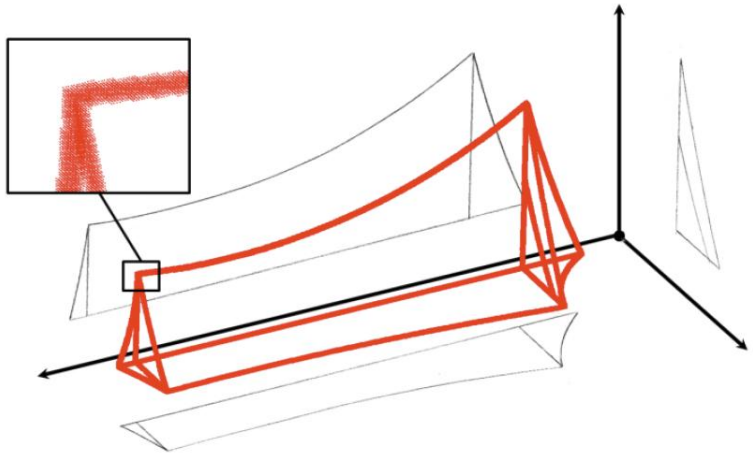


Figure A2: 3D image G (non zero voxels are plotted).

SEPARATING BRANCHES BY MEANS OF APPROXIMATE INTERSECTION DETECTION

6) Image *G* is converted into a binary image *T* ($T(i,j,k) = 1$ *if* $G(i,j,k) \neq 0$) which undergoes a morphological closing to remove possible small cavities. The resulting voxel cloud is transformed into a triangular mesh which is iteratively smoothed like shown in Figure A3 (laplacian smoothing is used).

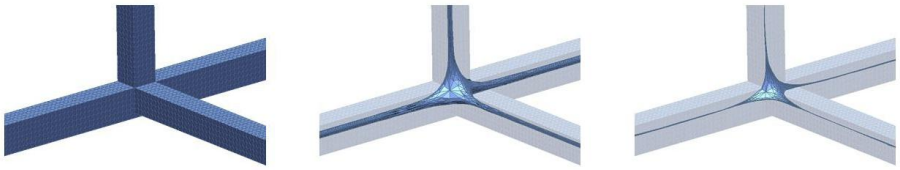


Figure A3: Laplacian smoothing process on the triangular mesh obtained from image T (detail of an intersection region).

7) The resulting triangular mesh is characterized by a large number of spikes (situated in the regions far from the intersections) and a few approximately equilateral triangles (situated in the regions where multiple “wires” intersect). In order to spot the approximate intersections, triangles’ form factor f is analyzed and triangles with $0.1 < f < 1$ are selected. The centroids ci of each connected group of such triangles are assumed to be a first guess for the intersection points.

8) For each centroid ci, the equation of a sphere Ei is computed so that it is centered on ci and has a radius proportional to the dilation kernel dimension. The spheres are assumed to represent the regions where the wires’ intersections lie. Accordingly, in order to separate the “branches” composing the wireframe-like voxel model, image T is “cut” by means of spheres Ei by setting $T(i,j) = 0$ for all the voxels lying inside any of the spheres.

9) The binary 3D image resulting from phase 8 is labeled according to an appositely developed memory efficient algorithm. For each labeled region, the corresponding one is extracted from image G and stored in a separate 3D image.

OBTAINING A 3D WIREFRAME MODEL MADE OF SPLINE CURVES, STARTING FROM EACH LABELED ENTITY OF 3D IMAGE G

10) Each separate branch is fitted by means of a spline curve according to a process based on the use of weighted least squares. Being the weights represented by the voxel values, the resulting spline curves tend to be well superimposed to the voxel cloud regions with voxel values ≈ 1; as explained in phase 5, these are the regions originated by well matching 2D projections.

11) once the fitting process is complete, final intersection points are re-computed by extending splines curves converging into a intersection region (determined in phase 8) along their tangent direction and determining mutual minimum distance point pi for each region (Figure A4). Since the intersection point cannot lie outside of the voxel cloud, pi is selected among the voxels belonging to the spherical region.

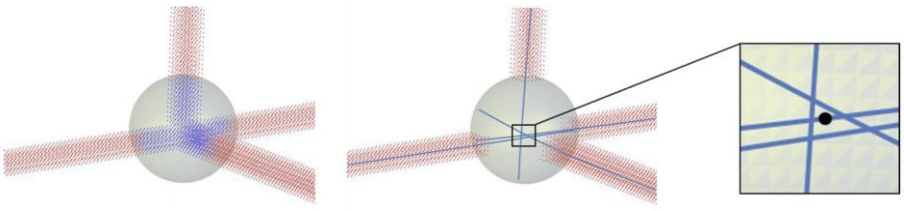


Figure A4: Cutting sphere (left); extended spline curves with detail view showing pi (right)

12) each branch is re-fitted using a spline curve with appropriate boundary condition so that its initial and final points coincide with the intersections found in phase 11.

13) the resulting set of spline curves is reassembled in a single model which represent the vector wireframe of the original object. The model can be translated into a DXF file.

14) the edges of the wireframe model are used as support entities to define a set of surface patches (Figure A5). This phase has been performed both manually (importing the DXF file in a surface modeling software environment and selecting edge cycles defining each single surface patch) and semi-automatically.

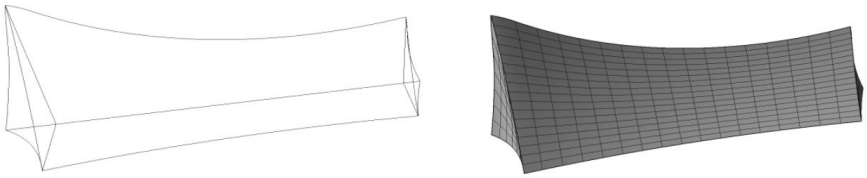


Figure A5: Reassembled set of spline curves (left) and final surfaces (right).

RESULTS AND DISCUSSION

The reconstruction methodology, implemented using Matlab® programming language (except for the surface creation phase, which has been carried out using Rhinoceros 4.0), has been applied on a number of case studies and proved to be quite robust and to require only a little interaction from the operator. Due to its working principle, based on voxel cloud elaboration and surface fitting procedure, the method provides a resulting 3D geometry which can be considered a draft version of the one achievable by means of a conventional 3D modeling process. However, the manual reconstruction of the vector wireframe proves to be considerably speeded up even if some reworking may be necessary for adding some kind of constraints (e.g. symmetry, tangency, etc.). Future work will be addressed to deal with this kind of issues and, possibly, to interface the developed procedure with a commercial CAD software package.

Indice

1 Introduzione 15

2 Stato dell'arte della ricerca accademica 31

2.1 Ricostruzione di poliedri a partire dal modello wireframe 33

2.1.1 Controllo dei dati di input 34
2.1.2 Ricerca di grafi planari 35
2.1.3 Calcolo dei 1-cycle e delle virtual face 36
2.1.4 Controllo delle intersezioni illegali tra virtual face 37
2.1.5 Determinazione dei 2-cycle e dei virtual block 37
2.1.6 Costruzione delle soluzioni 38

2.2 Ricostruzione del modello wireframe a partire da proiezioni ortogonali .. 39

2.2.1 Controllo dei dati di input 42
2.2.2 Costruzione di uno pseudo scheletro tridimensionale 42
2.2.3 Costruzione di uno pseudo wire frame 43

2.3 Sviluppi più recenti 43

2.3.1 Gestione delle curve coniche 44
2.3.2 Ricostruzione con approcci di tipo "human-like" 45

2.4 Ricostruzione mediante l'uso di template 50

3 Metodologia 57

3.1 Costruzione del modello raster 3D 57

3.1.1 Rimozione degli errori di orientazione causati dalla scansione 58
3.1.2 Identificazione delle viste e loro estrazione 60
3.1.3 Operazioni di image processing sulle viste estratte 61
3.1.4 Generazione dei modelli 3D 63

3.2 Estrazione della topologia del modello raster 3D 64

3.2.1 Metodo delle distanze 65
3.2.2 Metodo del gradiente 66
3.2.3 Metodo della soglia 68
3.2.4 Metodo della scheletonizzazione 3D 78
3.2.5 Metodo dello smoothing 78

3.3 Generazione dei vertici finali 88

3.3.1 Estrazione delle traiettorie 89
3.3.2 Posizionamento dei vertici 97

3.4 Generazione delle curve finali 98

3.5 Generazione delle patch di superficie 104

3.5.1 Estrazione dei cicli dal modello wireframe 3D e generazione delle superfici finali 105

4 Casi studio 117

5 Conclusioni 135

Elenco delle Figure 139

Bibliografia 147

1 Introduzione

Negli ultimi decenni la progettazione in tutte le sue mille sfaccettature ha ricevuto un importantissimo contributo dall'utilizzo del computer, con impatto enorme in termini di tempi, costi e qualità dei risultati raggiunti. La progettazione assistita dal calcolatore consente, infatti, di simulare il comportamento di un generico prodotto (del quale si dispone di un modello computazionale) da molti punti di vista (strutturale, estetico, ergonomico, etc.), riducendo drasticamente il numero di prototipi fisici e prove sperimentali necessarie per verificarne la corretta progettazione. Inoltre, grazie all'uso di strumenti software evoluti, si è ottenuta una notevole integrazione tra vari aspetti e fasi della progettazione stessa, fra cui il dimensionamento, la verifica, gli sudi dinamici, nonché l'ottimizzazione dei materiali, dei costi e delle prestazioni, fino a comprendere anche i processi stessi di fabbricazione.

E' evidente, quindi, che al giorno d'oggi i sistemi di computer-aided design (CAD) sono ritenuti fondamentali, anche perché, per essere competitivi sul mercato, occorre svolgere in modo estremamente rapido le fasi di sviluppo prodotto.

Sebbene in molti casi si possa considerare il computer come una sorta di punto di partenza del processo di sviluppo di nuovi prodotti, in molti altri ciò non può ancora avvenire, oppure avviene ma con procedure lente e difficoltose. Questo risulta evidente pensando ai casi in cui l'elemento stilistico assume un ruolo primario, per esempio per i prodotti di design. Per questo tipo di prodotti, spesso si parte ancora da schizzi e proiezioni a matita (il cosiddetto bozzetto) o addirittura da modelli fisici (prototipi) realizzati con vari materiali come ad esempio gesso, resina, cera, legno, ecc. che poi devono essere, in qualche modo, trasferiti su un sistema CAD per poter iniziare la successive fasi dello sviluppo tra cui l'attività di progettazione vera e propria.

Anche se i sistemi CAD tridimensionali sono largamente utilizzati nel campo del cosiddetto "design" (nell'accezione italiana del termine), convenzionalmente lo schizzo a mano libera è uno dei metodi preferiti nelle prime fasi del design stilistico. Sia i modellatori CAD che i più recenti sistemi di CAS scoraggiano la creatività del designer poiché risultano inadeguati per la rapida realizzazione dell'idea stilistica.

Nella primissime fasi dell'ideazione di un nuovo prodotto i designer producono, infatti, un ricco set di schizzi concettuali per comunicare e sviluppare le proprie idee.
E' possibile fornire una spiegazione del perché il disegno di stile su carta sia ancora la forma creativa preferita dai designer; tale spiegazione può essere articolata in tre punti principali che caratterizzano e distinguono lo strumento del disegno di stile a mano libera rispetto ad altri strumenti, sempre più basati sulle nuove tecnologie, per la creatività ed il design.
Dal momento in cui un bozzetto viene concepito, il designer ha la capacità di monitorarne costantemente la sua evoluzione, tratto dopo tratto [1]. Questa sua dote gli permette di reinterpretare l'immagine visiva dello schizzo confrontandolo continuamente con il suo concetto mentale. Il designer può apportare correzioni al disegno per avvicinarlo alla sua raffigurazione mentale, può anche utilizzare queste differenze per aggiornare l'idea su cui sta lavorando cercando nuovi concetti oppure semplicemente completando il suo schema mentale ovvero definendo caratteristiche ed aspetti del bozzetto che non aveva ancora affrontato.
Il secondo aspetto fondamentale del disegno a mano libera si identifica nelle capacità di ridefinizione continua dello schizzo tipiche del designer. Tali capacità sono note in letteratura come "oversketching" [2], "re-sketching", "re-drawing", "overtracing" [1], "scribbling", e "nervous hand" [3]. Con questi termini i diversi autori intendono descrivere la capacità del designer di utilizzare la sua matita per enfatizzare aspetti prioritari del disegno facendone scivolare in secondo piano altri di minore importanza; sebbene la questione possa apparire di poco significato tecnico, tali abilità permettono al designer di costruirsi un modello mentale secondo il quale tratti di diversa natura contengono diverse informazioni; ovvero importanti "feedback" che permettono all'artista di avanzare nella realizzazione del disegno.
Il terzo ed ultimo punto cardine favorevole all'impiego del disegno a mano libera è quello che in letteratura viene chiamato "incremental refinement" e che racchiude in se il concetto di come ridefinizione e feedback concorrono assieme allo sviluppo di un'idea [1,4]. Dal macroscopico al microscopico, con il progredire del disegno verso la sua configurazione finale si attraversano molte fasi intermedie sulle quali il designer lavora per definire al meglio il concetto che sta esprimendo tramite la sua matita.
Alle considerazioni di cui sopra se ne aggiungono altre per cui la tradizionale pratica dello schizzo 2D è da sempre la più attrattiva nella fase di ideazione del concetto e nella successiva opera di esplicitazione grafica delle possibili varianti.
In primo luogo viviamo in un periodo storico ancora troppo vicino alla nascita dei moderni strumenti CAD; ad oggi questi costituiscono uno strumento volto alla soluzione di problemi prettamente tecnici e sono poco orientati ad assistere la fase iniziale legata all'ideazione e al primo sviluppo di un nuovo prodotto.

In secondo luogo l'operazione stessa di "gettare" idee su di un foglio carta è di gran lunga la soluzione che permette di esprimere al massimo la creatività del designer. Forzare il designer a seguire una serie di vincoli dettati dalle esigenze tecniche di un comune strumento CAD significherebbe limitarne le potenzialità con conseguente degrado del suo apporto creativo allo sviluppo dell'idea; per molti designer sviluppare le proprie idee direttamente in un ambiente CAD 3D risulterebbe notevolmente più complicato che lavorare su di un buon disegno bidimensionale in proiezioni ortogonali [5].

Tuttavia ottenere un modello CAD tridimensionale da questi schizzi, spesso creati a mano libera, è un processo tanto laborioso con gli strumenti attualmente disponibili quanto necessario per restare al tempo con le moderne tecniche di progettazione e di produzione. Anche nelle fasi più avanzate del design, quando il bozzetto ha assunto una sua ben definita geometria e mostra forti analogie con i disegni tecnici, la difficoltà di una semplice e rapida ricostruzione si manifesta in maniera evidente. Eppure, giunti a questa fase (Figura 1.1) è essenziale valutare la qualità del design stesso (selezione delle alternative).

Tali valutazioni sono da intendersi necessarie sia da un punto di vista estetico (per mezzo, ad esempio, di rendering fotorealistici) che dal punto di vista più strettamente ingegneristico; ovvero è necessario comprendere se vi sono, ed in tal caso quali sono, aspetti che impediscono o complicano la fattibilità del progetto, in altre parole le criticità maggiori in prospettiva di una futura pianificazione del processo di ingegnerizzazione e produzione. Risulta inoltre spesso necessario valutare, anche mediante l'ausilio di modelli CAD tridimensionali la corretta funzionalità del prodotto: è noto, infatti, che un intervento correttivo è assai meno invasivo se effettuato nelle primissime fasi di sviluppo del prodotto.

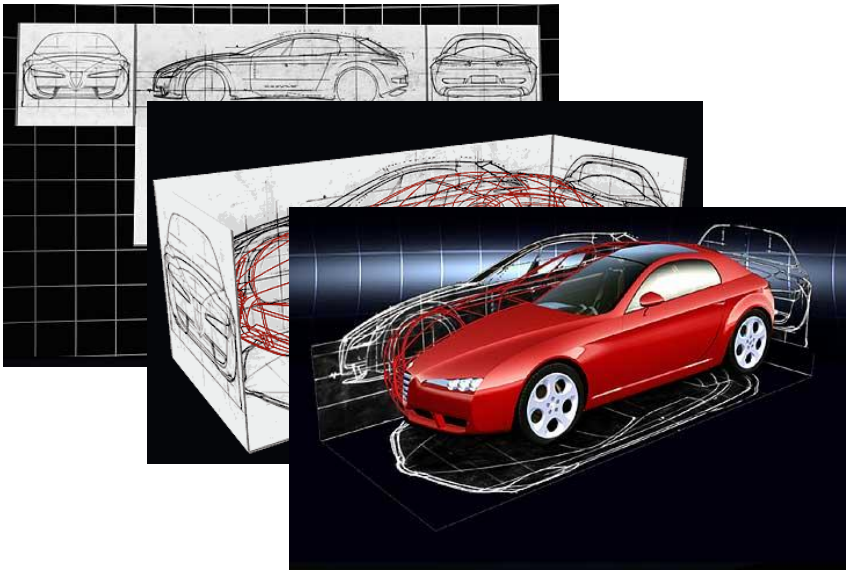


Figura 1.1 – Descrizione grafica del processo di ricostruzione 3D di un disegno tecnico di stile su carta

Nel caso (abbastanza frequente) in cui il designer produca una rappresentazione della propria idea impiegando prototipi fisici le tecnologie di scansione 3D (e più in generale le tecniche di Reverse Engineering) rappresentano un potente strumento per passare in tempi ragionevoli al modello CAD in modo automatico o semiautomatico. Attualmente i sistemi di acquisizione 3D hanno raggiunto una sofisticazione tale da riprodurre le geometrie di oggetti complessi con elevata precisione e ripetibilità; i software, dal canto loro, mettono continuamente a disposizione nuove funzionalità per velocizzare e migliorare il processo di ricostruzione delle superfici a partire dalle nuvole di punti acquisite.
Ben più complesso è il caso in cui, invece, si voglia prendere in considerazione una qualunque immagine 2D (il disegno tecnico di stile appunto) e trasformarla in un modello 3D. La fase di conversione utilizzata ad oggi risulta del tutto manuale e coinvolge in modo diretto, quando possibile, sia il disegnatore CAD che lo stilista. Questa risulta, pertanto, estremamente dispendiosa in termini di tempo rappresentando una delle fasi più delicate del processo di definizione di un nuovo prodotto.
Un approccio per risolvere queste problematiche, limitatamente al disegno meccanico, cioè a oggetti riproducibili con un insieme di primitive e/o features, è stato proposto all'interno di software commerciali quali “Instant 3D” di Solidworks (Dassault Systèmes SolidWorks® Corp.) ed il tool "2D to 3D" di Inventor (Autodesk®). Tuttavia tali strumenti (Figure da 1.2 a 1.7) sono stati

specificatamente concepiti per assistere la ricostruzione manuale di semplici geometrie; risultano pertanto di scarsa efficacia nella gestione di modelli caratterizzati da geometrie free-form.

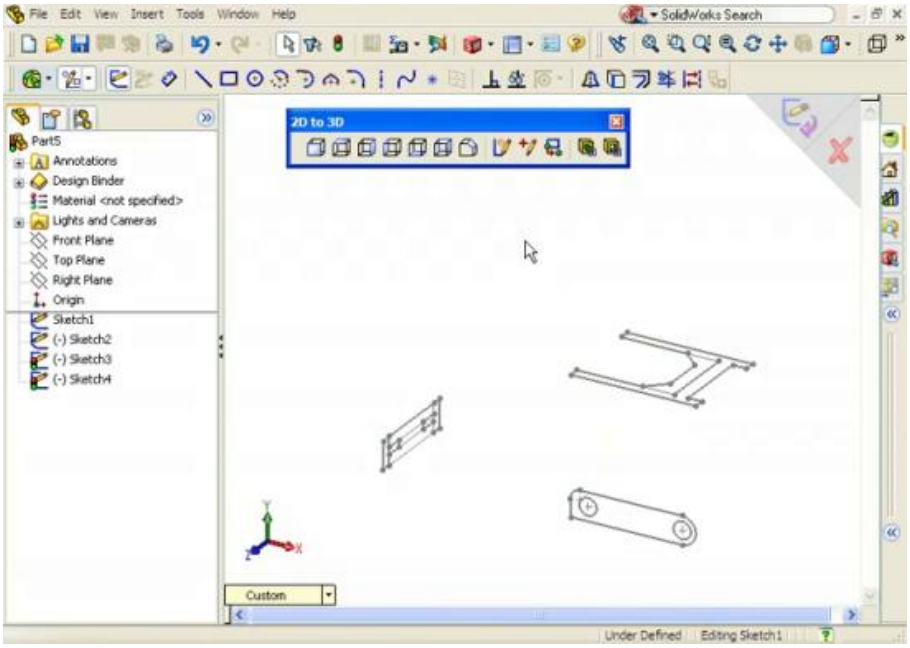


Figura 1.2 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software SolidWorks®: posizionamento relativo delle viste bidimensionali nello spazio di lavoro

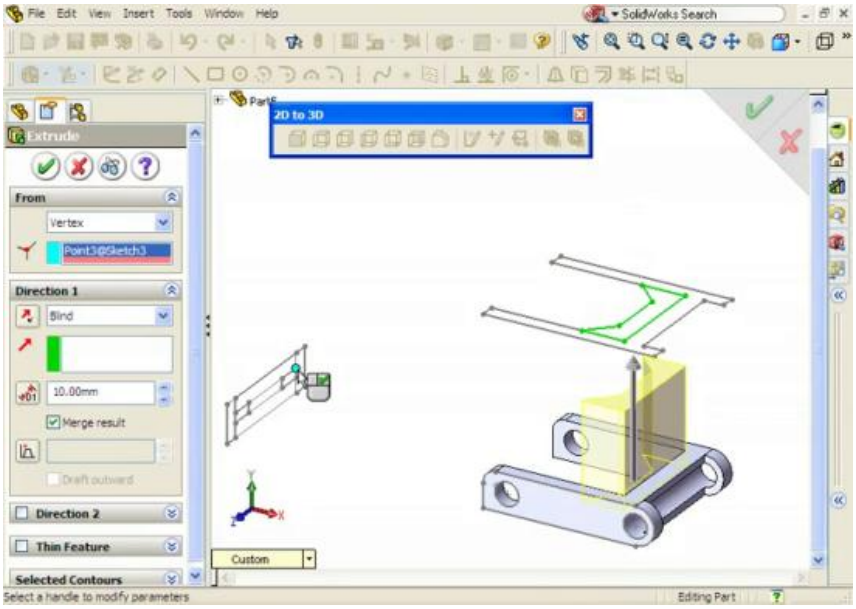


Figura 1.3 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software SolidWorks®: creazione del modello 3D mediante utilizzo delle informazioni contenute nelle viste bidimensionali.

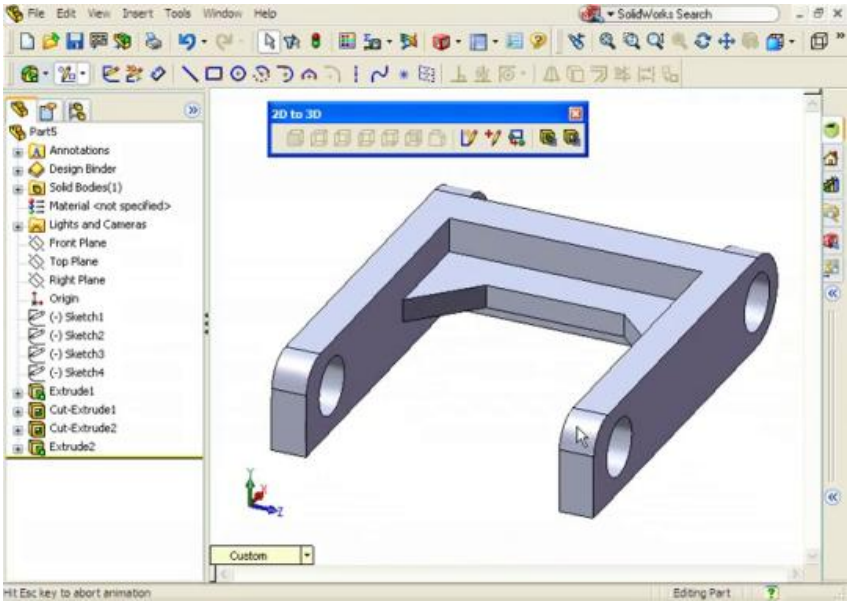


Figura 1.4 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software SolidWorks®: modello tridimensionale risultante dal lavoro di traduzione delle viste bidimensionali

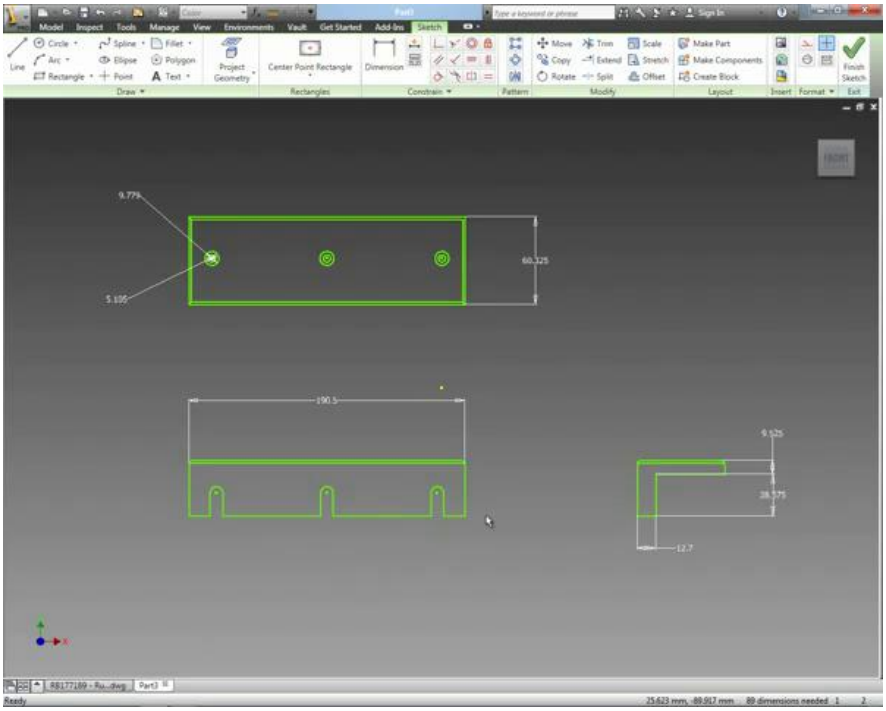


Figura 1.5 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software Inventor®: viste iniziali tipiche del disegno 2D.

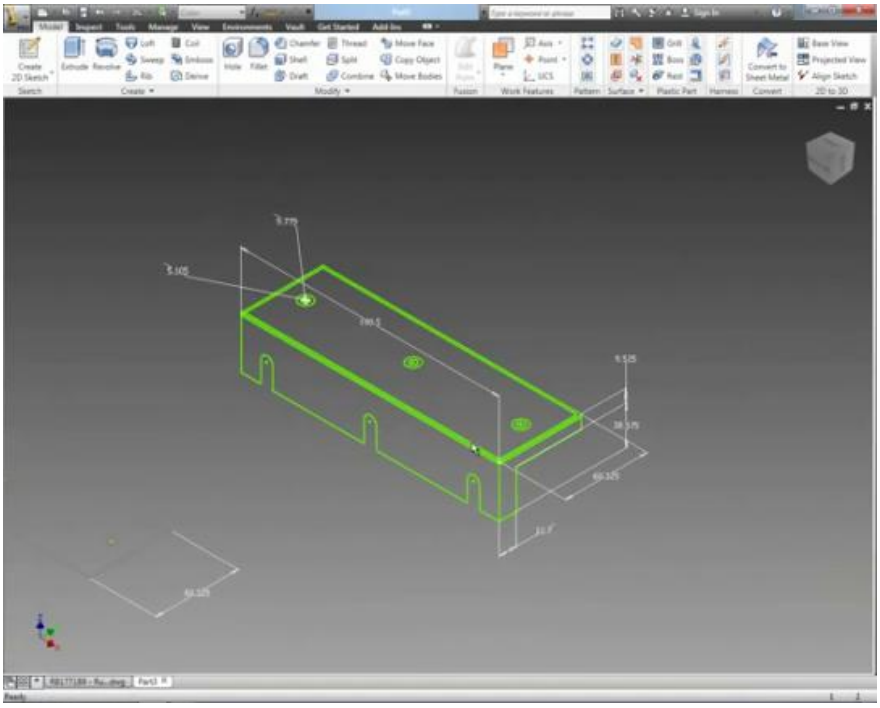


Figura 1.6 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software Inventor®: posizionamento relativo delle viste bidimensionali nello spazio di lavoro.

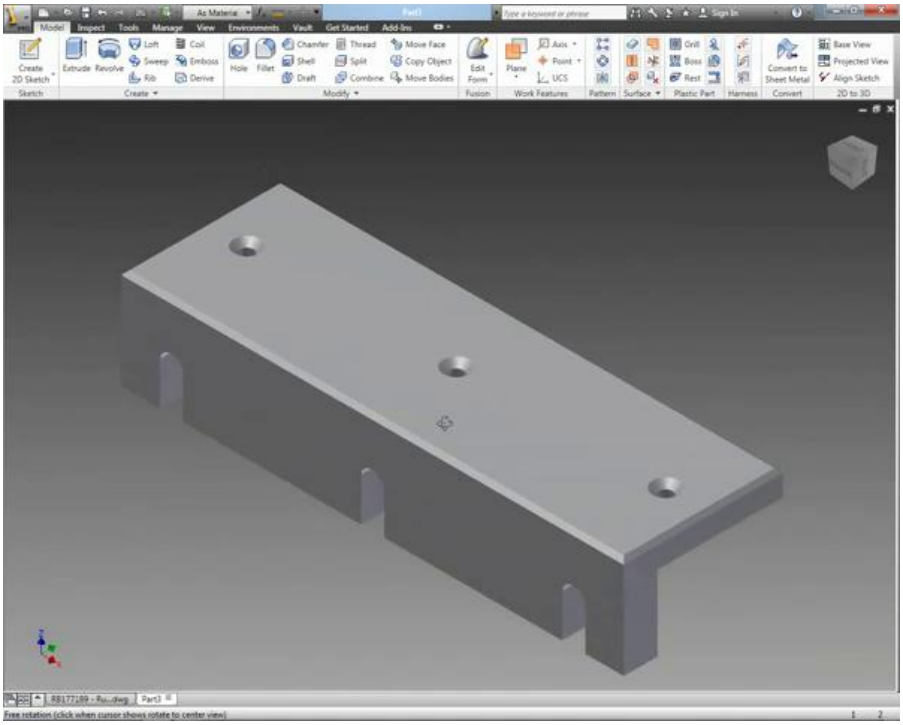


Figura 1.7 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software Inventor®: modello tridimensionale risultante dal lavoro di traduzione delle viste bidimensionali.

Tali sistemi permettono una ricostruzione tridimensionale virtuale di componenti a partire da file bidimensionali (ad esempio DWG o DXF) realizzati in ambiente CAD 2D. La ricostruzione avviene in maniera completamente manuale usando le viste bidimensionali come guida per le tipiche operazioni di disegno 3D quali estrusioni, rivoluzioni, ecc.

Di seguito, nelle Figure da 1.8 a 1.12, si descrive un percorso tipico della traduzione del disegno bidimensionale in modello tridimensionale. Nello specifico l'operazione è condotta utilizzando il software CATIA®.

Quello che si vuole sottolineare con questo esempio è la completa manualità di esecuzione; conseguentemente tale metodologia risulta di lenta applicazione e di forte rilevanza in termini economici. Un aspetto ulteriore riguarda la possibilità di interpretazioni soggettive che vincolano la bontà del risultato finale alle capacità del tecnico addetto alla ricostruzione ed alla sua continua collaborazione con il designer durante le fasi della ricostruzione.

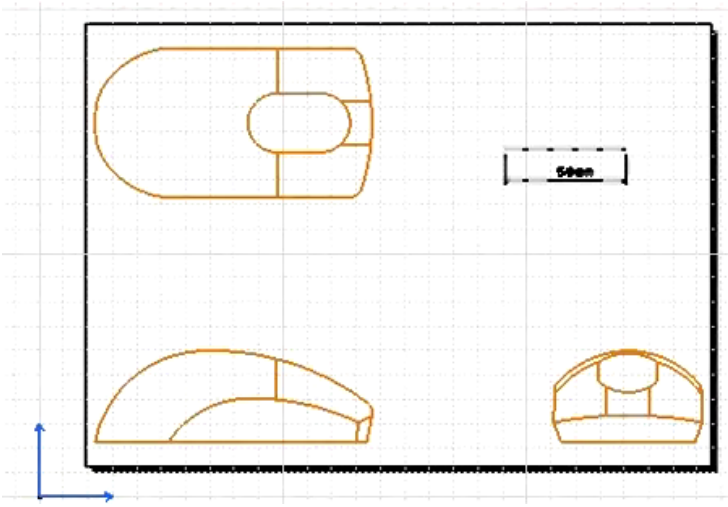


Figura 1.8 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: viste iniziali.

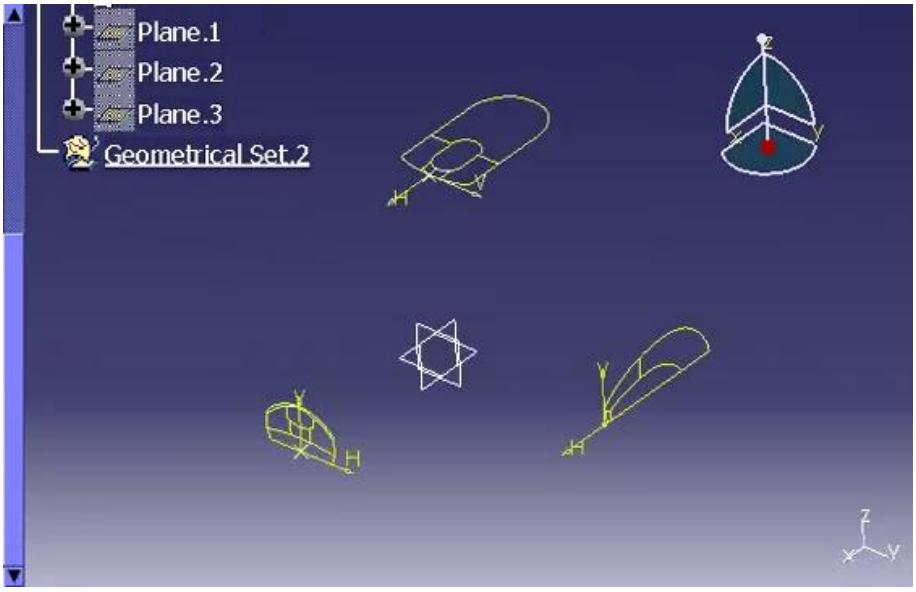


Figura 1.9 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: posizionamento relativo delle viste iniziali nello spazio di lavoro.

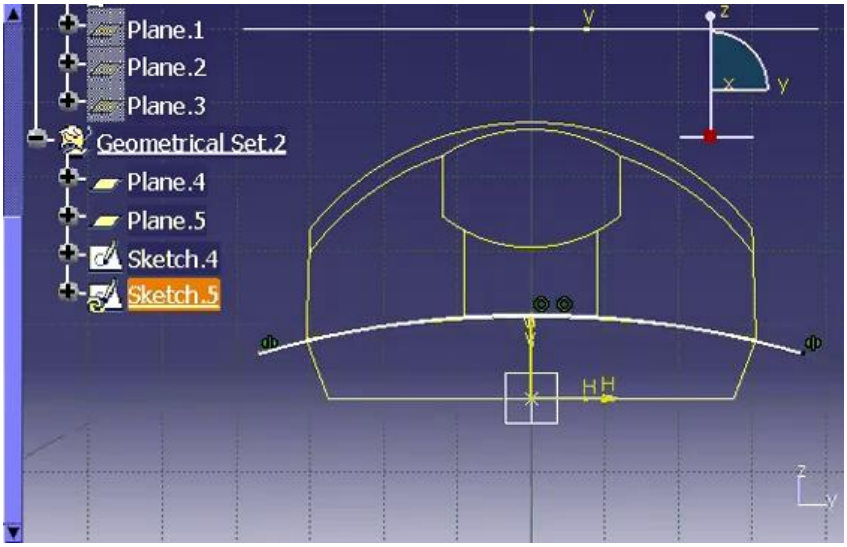


Figura 1.10 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: tracciamento manuale delle curve di stile sulle proiezioni bidimensionali di partenza.

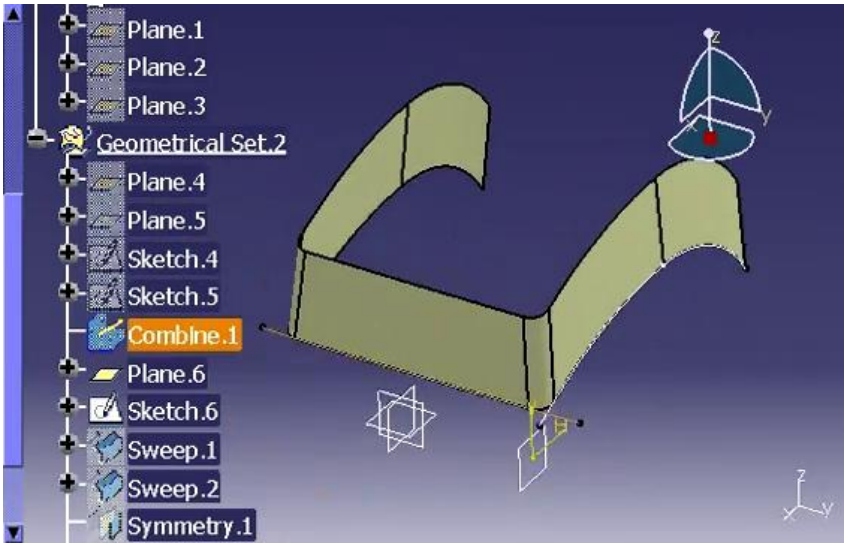


Figura 1.11 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: definizione delle patch di superfice giacenti sulle curve di stile estratte al passaggio precedente.

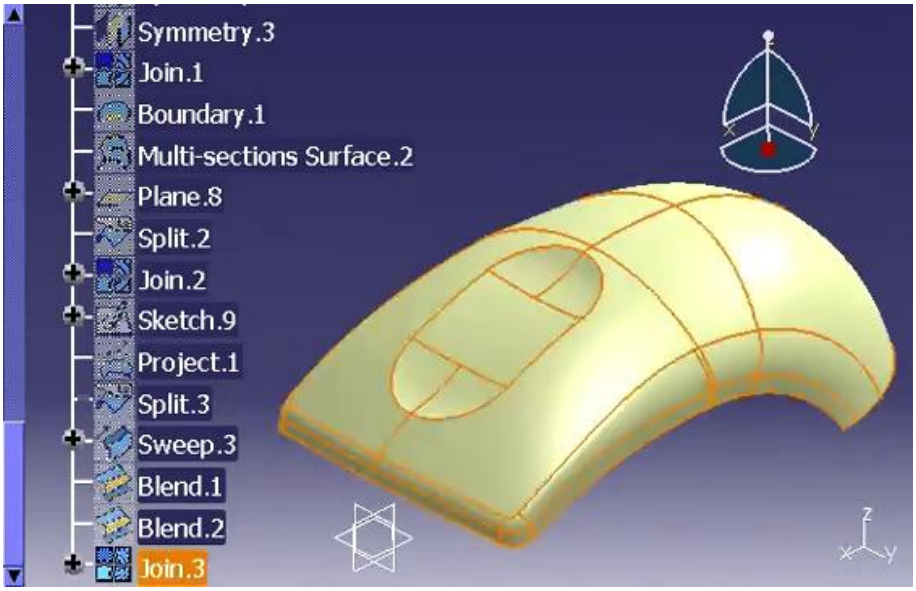


Figura 1.12 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: modello tridimensionale ottenuto al termine della ricostruzione.

Detti sistemi, pertanto, non consentono una rapida ricostruzione dei modelli 3D; a maggior ragione, dal momento che la ricostruzione avviene da rappresentazioni 2D "esatte" (vettoriali e dimensionalmente corrette), tali software non si rivelano adatti alla ricostruzione 3D a partire da bozzetti o disegni eseguiti a mano, anche se in proiezione ortogonale. Questi ultimi sono affetti da inevitabili errori ed approssimazioni di rappresentazione e, spesso, caratterizzati da ulteriori elementi informativi quali ad esempio ombreggiature e texture.

Per quanto riguarda il settore della ricerca scientifica (cfr. capitolo 2), molti approcci proposti in letteratura affrontano il problema della ricostruzione 3D con riferimento al disegno meccanico; questa tipologia di disegni, però, differisce profondamente da quelli stilistici per una serie di aspetti (presenza di spigoli vivi, facce piane e superfici primitive in genere nel caso di disegni meccanici, presenza di superfici raccordate e linee curve free-form nel caso di disegni di stile). Sulla base di quanto detto, le metodologie sviluppate per la prima tipologia di disegni, sono applicabili con notevoli limitazioni al campo del disegno stilistico. Esistono in letteratura anche alcuni studi volti alla ricostruzione da disegni di stile; parte di essi sono basati su template per la ricostruzione di particolari famiglie di prodotti [6], altri sono incentrati sulla sostituzione della carta con nuove tecnologie come le

tavolette grafiche [7], altri ancora basati su principi potenzialmente rivoluzionari come la realtà aumentata. Tuttavia nessuno di questi è in grado di fornire un approccio "generico" alla ricostruzione ponendo al designer vincoli più o meno importanti che ne limitano le capacità di espressione artistica.
Ad oggi, pertanto, è possibile concludere che non esiste uno strumento automatico o semiautomatico in grado di individuare le linee e le superfici di stile di un bozzetto o di un disegno a mano, seppure rappresentato in proiezione ortogonale, ed in grado di ricostruire tramite queste un modello CAD tridimensionale; eppure l'esigenza di un sistema di questo tipo è particolarmente sentita proprio nel nostro paese in quanto la produzione di oggetti di design è un elemento essenziale e distintivo del *Made in Italy*.
Evidentemente, come già accennato, lo sviluppo dei moderni sistemi CAD tridimensionali gioca un peso sempre maggiore nella pianificazione dello sviluppo di un nuovo prodotto. D'altro canto, anche per i motivi prima menzionati, l'utilizzo di bozzetti di stile ricavati mediante disegno a mano libera costituisce, e costituirà ancora in futuro, la metodologia più diffusa nelle prime fasi di sviluppo dell'idea.
Questo stato delle cose implica necessariamente che, ad un certo punto del processo di sviluppo del prodotto, il disegno a mano elaborato dal designer venga tradotto in geometrie tridimensionali funzionali al loro impiego negli ambienti CAD 3D. Il collocamento di questo evento, che viene solitamente identificato come il passaggio di consegne tra il designer ed il progettista, all'interno della "timeline" di sviluppo del prodotto rappresenta una scelta strategica.
La traduzione del modello 2D in un modello 3D avviene quindi nel momento in cui il designer ha definito le proprie specifiche sviluppando un elaborato finale che presenta molte analogie con un vero e proprio disegno tecnico pur essendo ricavato manualmente su carta (Figura 1.13).

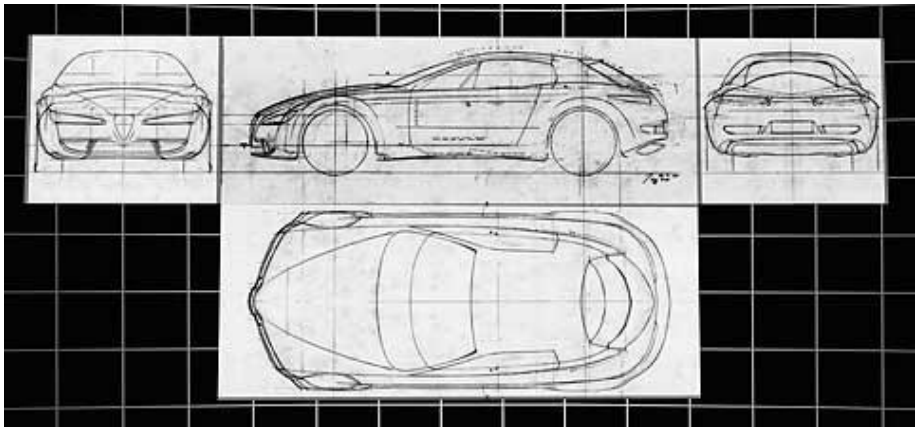


Figura 1.13 – Disegno di stile in stato avanzato di elaborazione

Il problema della traduzione in formato tridimensionale di un elaborato di questo tipo presenta alcune caratteristiche, almeno dal punto di vista concettuale, proprie dei problemi che si affrontano nel campo della cosiddetta “3D reconstruction”, espressione con la quale si intende identificare il problema della ricostruzione tridimensionale di un oggetto a partire da proiezioni ortogonali in formato vettoriale, talvolta comprensive di informazioni addizionali quali ad esempio annotazioni e sezioni.
Proprio con riferimento alla *3D reconstruction,* negli ultimi decenni ci sono stati diversi tentativi di individuare una soluzione generale al problema (cfr. capitolo 2). La nascita e l’enorme sviluppo dei sistemi CAD ha inevitabilmente provocato il moltiplicarsi degli studi sull’argomento e la nascita di varie correnti di sviluppo che hanno portato alla pubblicazione di numerosi tecniche per la ricostruzione.
I lavori presentati nel corso degli anni si contraddistinguono fortemente per il loro orientamento al mondo della progettazione meccanica. Solo più tardi sono state proposte alcune metodologie che affrontano (ma solo parzialmente) il problema della ricostruzione da disegni di stile, ossia di tipo free form.
Eppure, ad oggi, nessun approccio generalizzato è ancora stato presentato per la soluzione della traduzione automatica di disegni di stile bidimensionali in modelli CAD tridimensionali.
Il presente lavoro ha avuto fin dal suo principio lo scopo di sopperire a questa lacuna; in particolare posando lo sguardo su campi di ricerca affini con lo scopo di produrre una serie di tecniche e di metodi, generalmente validi, per la traduzione automatica di disegni tecnici di stile in geometrie CAD tridimensionali.

2 Stato dell'arte della ricerca accademica

Come accennato nel capitolo precedente, il lavoro svolto in seno al Dottorato è stato basato su un'accurata indagine indirizzata a campi di ricerca affini. Tale indagine non poteva che iniziare dal filone di ricerca che affronta il problema noto come "3D reconstruction". Per molti versi, come già detto, tale campo di ricerca può essere preso ad esempio per l'elaborazione di tecniche e metodi applicabili specificatamente ai disegni di stile.

Le tecniche ad oggi sviluppate per affrontare il problema della "3D reconstruction" consentono, infatti, la traduzione di proiezioni ortogonali esatte di tipo vettoriale in modelli CAD tridimensionali con procedimenti più o meno automatici. Tuttavia, proprio in questa definizione è possibile cogliere l'aspetto principale che impedisce lo sfruttamento diretto di tali tecniche nella risoluzione del problema affrontato in questa attività di Dottorato: i disegni di stile, pur nella loro forma più evoluta, affine a quella caratteristica di un disegno tecnico, per loro natura non possono fornire una rappresentazione esatta di tipo vettoriale.

Il dato in input del problema qui trattato differisce, pertanto, da quello disponibile nei problemi classici di "3D reconstruction". La sua natura di tipo raster unita alla ovvia mancanza di corrispondenze "esatte" tra le diverse proiezioni del disegno di stile origina una fatale mancanza di dati che ne preclude l'impiego con le tecniche di ricostruzione sopra citate.

Anche tramite l'aiuto dei sempre più diffusi strumenti software per la vettorializzazione di immagini raster come ad esempio WinTopo, Scan2CAD o VectorMagic risulta poco percorribile l'ipotesi di estrarre informazioni esatte di tipo vettoriale utili allo sfruttamento delle suddette tecniche di ricostruzione.

Tali strumenti, pur assolvendo al loro scopo in modo sempre più efficace, non consentono certo di ricreare un set di proiezioni ortogonali dove le corrispondenze tra le varie viste possano essere determinate in maniera esatta. E'

infatti semplice da immaginare come sia concettualmente impossibile per un disegno di stile, realizzato manualmente, avere corrispondenze "esatte" tra le varie viste. Come ulteriore grave criticità si consideri anche la necessità di operare con geometrie complesse, di natura free-form, che sono alla base del disegno di stile. Le tecniche più avanzate di 3D reconstruction permettono la gestione di semplici curve al massimo di tipo conico.

Mantenendo ben chiara la differenza tra il problema che si intende affrontare in questa sede e quello che si propongono di risolvere le tecniche di "3D reconstruction", quest'ultimo resta di gran lunga il campo di ricerca con la più elevata affinità, dal quale poter trarre spunti ed idee per la soluzione della ricostruzione tridimensionale a partire da disegni tecnici di stile.

Allo scopo di individuare i contributi scientifici di maggiore rilievo e pertinenza con l'argomento di questo lavoro, sono stati analizzati una serie di studi condotti a partire dai primi anni '70. Alcuni lavori, selezionati tra i molti esaminati, vengono presentati con maggiore attenzione al termine di questo capitolo introduttivo. lo studio approfondito di questi lavori è necessario e dovuto per far meglio comprendere come sia effettivamente impossibile la gestione dei disegni di stile con le stesse tecniche adottate per la ricostruzione di disegni tecnici (di particolari meccanici), caratterizzati da proiezioni ortogonali esatte.

Iniziamo, dunque, a ripercorrere le tappe fondamentali della ricerca nel campo della "3D reconstruction".

Nel 1973 Idesawa [8] propone uno dei primi modelli di ricostruzione basato su schema b-rep (boundary representation), introducendo un algoritmo per la gestione delle tre viste ortogonali e trattando esclusivamente spigoli rettilinei e facce poligonali (in altre parole, solidi poliedrici); Idesawa evidenzia alcune lacune, da addebitare sia alla scarsa potenza computazionale dei sistemi informatici dell'epoca, sia a criticità interne all'algoritmo stesso; infine, l'autore introduce le numerose problematiche connesse all'utilizzo di tutte le informazioni ausiliarie presenti nel disegno (quali viste in sezione, quotature, annotazioni, …) che sarebbero state riprese ed affrontate negli anni successivi.

Trascorrono tre anni e Lafue, nel 1976, pubblica un nuovo studio sulla ricostruzione tridimensionale [9]. In questo caso l'attenzione dell'autore si focalizza sulle inevitabili ambiguità che possono nascere in una rappresentazione bidimensionale. Viene quindi proposto un approccio euristico per la soluzione di tali ambiguità, prevedendo un intervento esterno, da parte dell'operatore, nel caso in cui l'algoritmo non riesca ad avanzare.

Nel corso dei primi anni '80 vengono pubblicati due studi [10,11], da parte di Wesley e Markovsky, che costituiranno nel tempo il più valido riferimento bibliografico per la quasi totalità di lavori finora pubblicati. Il modello proposto da Wesley e da Markowsky verrà quindi adottato quale linea guida; i successivi lavori

tenderanno, per lo più, ad ampliarne gli orizzonti, soprattutto alla luce di una crescita esponenziale di potenza computazionale dei calcolatori.

La pubblicazione di questi due lavori costituisce un punto di svolta per quanto riguarda l'attenzione del mondo della ricerca a queste tematiche; se fino al 1980 erano stati condotti soltanto pochi studi sulla materia, da li a poco la bibliografia relativa alla ricostruzione tridimensionale da proiezioni ortogonali si sarebbe arricchita molto rapidamente [12–27], fino ad arrivare a centinaia di pubblicazioni soltanto sui giornali più importanti dedicati al disegno tecnico e all'informatica grafica.

La strada segnata da Wesley e Markowsky viene percorsa da vari autori con particolare attenzione all'efficienza dei modelli proposti, cercando sempre migliori prestazioni in termini di velocità e robustezza nei vari algoritmi. Proprio a causa della forte rilevanza nel contesto scientifico, di seguito si descrive brevemente proprio l'approccio di Wesley e Markowsky.

Si fa notare che i due lavori vengono proposti in ordine di pubblicazione mentre dal punto di vista algoritmico vale l'esatto contrario, la parte presentata per prima si applica al risultato ottenuto mediante le tecniche descritte nel secondo lavoro dei due autori.

2.1 Ricostruzione di poliedri a partire dal modello wireframe

Dato un qualsiasi oggetto poliedrico, il suo modello wire-frame è costituito dall'insieme dei suoi spigoli e dei suoi vertici. Nel lavoro di Wesley e Markowsky, intitolato "Fleshing out wire frames" [10], viene presentato un algoritmo che consente di ricostruire tutti i possibili "oggetti" aventi una determinata rappresentazione wire-frame. Nella forma in cui viene qui presentato (e pubblicato dagli autori), il modello è ristretto ai soli oggetti aventi spigoli rettilinei e facce planari (poliedri). La natura topologica dell'algoritmo non ne preclude tuttavia l'applicazione ad oggetti di più complessa geometria. Un'osservazione necessaria è legata alla probabilità di incorrere in soluzioni multiple, questo poiché una descrizione di un solido tramite i suoi spigoli porta con sé una chiara perdita di informazioni circa la sua vera geometria. L'utilizzo di questo algoritmo non può, ovviamente, sopperire a questa lacuna ma può comunque essere di grande aiuto; il processo metodico attuato dall'algoritmo di Wesley e Markowsky permette infatti la ricostruzione dell'intero insieme di oggetti caratterizzati dal modello wire frame di partenza, annullando la possibilità di scartare a priori la soluzione corretta. Come in molti problemi di natura geometrica, i casi semplici sono di banale risoluzione mentre aumentando la complessità geometrica aumenta la difficoltà di generare una corretta interpretazione; alcuni casi patologici riguardano vertici e spigoli a contatto con

facce oppure facce con normale opposta con uno spigolo a contatto. Nel corso dell'analisi condotta nel terzo capitolo verranno richiamati ed approfonditi alcuni passaggi di questo studio; per una corretta comprensione sono state aggiunte in appendice "A" alcune semplici nozioni di topologia.

Le fasi in cui si può suddividere l'algoritmo sono le seguenti:

1. controllo dei dati di input;
2. ricerca di grafi planari;
3. determinazione dei 1-cycle e delle virtual face;
4. controllo delle intersezioni illegali tra virtual face;
5. determinazione dei 2-cycle e dei virtual block;
6. costruzione delle soluzioni.

Nel seguito ciascuna fase è brevemente descritta in modo da chiarire il funzionamento di base dell'algoritmo.

2.1.1 Controllo dei dati di input

Si assume che i dati di ingresso costituiscano una valida rappresentazione wire frame e che siano disponibili sotto forma di due insiemi, nodi (vertici) e archi (spigoli). In questa fase vengono preposti opportuni controlli allo scopo di eliminare eventuali errori. Le tipologie di controllo devono essere adottate secondo due criteri principali. Il primo riguarda la fonte di acquisizione dei dati ed opera nell'ottica di poter prevedere le tipologie di errori più probabili; il secondo valuta il rapporto costi benefici dell'eventuale controllo, facendo sì che vengano introdotti sia quei controlli che, a fronte di un onere computazionale ridotto, escludono o limitano pericoli di malfunzionamento del modello, sia quelli che, pur comportando costi macchina consistenti, proteggono l'algoritmo da elevati pericoli di fallimento. Tra i vari test eseguibili sulla base dati ne esistono alcuni che, proprio grazie al loro rapporto costi benefici estremamente favorevole, sono fortemente consigliati; ad esempio il controllo sulla presenza di vertici o spigoli sovrapposti è un indice dell'esistenza nella struttura di dati ridondanti che, oltre a costituire informazioni inutili, possono creare pericoli per il corretto svolgimento dell'algoritmo.

2.1.2 Ricerca di grafi planari

Partendo dalla rappresentazione wire frame dell'oggetto, ogni piano geometrico contenente due o più spigoli intersecanti viene calcolato e archiviato. Per ciascuno vengono calcolate una delle due normali e una lista degli spigoli contenuti sul piano; da quest'ultima viene poi generato un grafo orientato. Ad ogni vertice appartenente al piano viene infine associata una lista degli spigoli giacenti su tale piano e per i quali il vertice costituisce un punto finale; tale lista viene creata seguendo la regola della mano destra rispetto alla normale del piano precedentemente adottata. In figura 2.1 si riportano, a titolo d'esempio, due piani; per quanto riguarda il "piano 1" viene mostrata anche la procedura di enumerazione degli spigoli attorno ad un generico vertice "v". Ovviamente la lista può essere indifferentemente "e1-e2-e3", "e2-e3-e1" o "e3-e1-e2", ciò che conta è l'ordine.

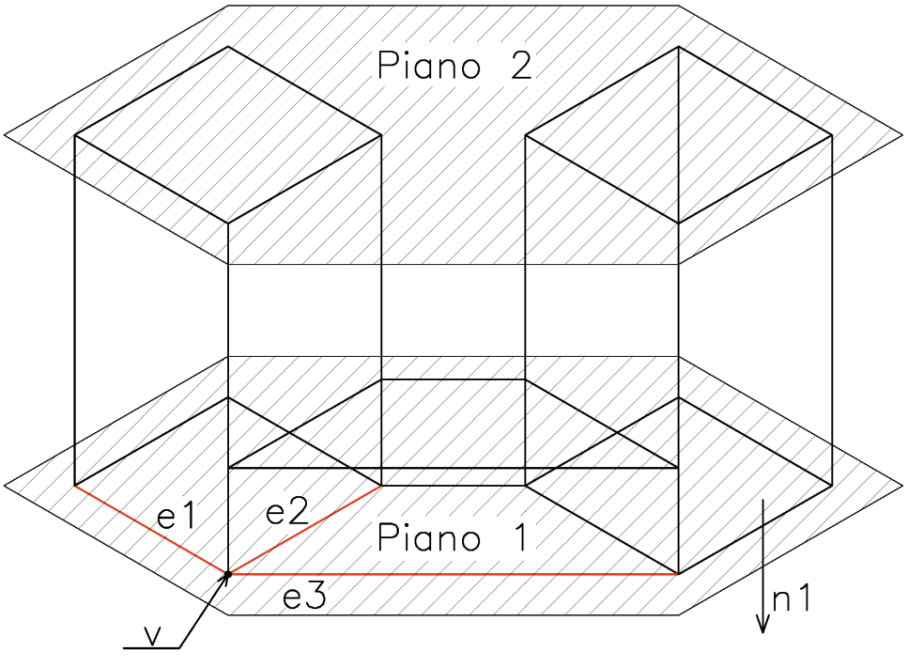


Figura 2.1 – Cicli planari

Il risultato di questo passaggio consiste in:

1. una lista di piani caratterizzati ciascuno da una normale e dalla distanza dall'origine lungo tale vettore;

2. per ciascun piano, un grafo contenente informazioni circa gli spigoli e i vertici giacenti su di esso.

2.1.3 Calcolo dei 1-cycle e delle virtual face

Dato l'insieme dei grafi planari, l'algoritmo procede esaminando ciascun grafo e determinando i cicli in essi contenuti; questi vengono etichettati come virtual face e vanno a costituire l'insieme dal quale saranno estratte le facce della soluzione finale. Lo schema logico di ricerca dei cicli prevede che ciascun arco del grafo in esame venga percorso in entrambi i sensi, stabilendo che, una volta determinato il ciclo, la regione associata agli archi percorsi dipenda dal verso di tale percorrenza; se il ciclo viene percorso in senso antiorario, si considera appartenente alla regione di superficie interna ad esso, viceversa, si considera appartenente alla regione esterna. Durante la generazione del singolo ciclo ciascun arco può essere percorso in una sola direzione; il mancato rispetto di questa regola rivela la presenza di un arco non ammissibile detto bridge e comporta la sua esclusione dall'insieme degli archi. Se tra le due percorrenze del bridge sono stati percorsi altri archi, questi costituiscono un ciclo che, come tale, viene archiviato tra le virtual face; per quanto riguarda gli archi attraversati a monte del primo passaggio sul bridge, l'informazione relativa alla loro percorrenza verrà eliminata. Ad esempio, con riferimento alla figura 2.1, supponendo che la ricerca di cicli parta da "e1" si ottiene la serie "e1-e2-e4-e5-e6-e7-e8-e2" che contiene due volte lo spigolo "e2"; quest'ultimo rappresenta un bridge e viene quindi eliminato, il ciclo "e4-e5-e6-e7-e8" viene archiviato e lo spigolo "e1" riportato alla precedente configurazione.

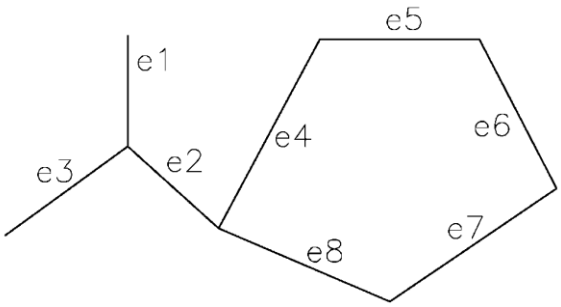


Figura 2.2 –Bridge all'interno di cicli planari

2.1.4 Controllo delle intersezioni illegali tra virtual face

Il corretto contatto tra due virtual face è quello riportato in figura 2.3; due facce possono infatti essere collegate esclusivamente da uno spigolo in comune ai loro contorni; inoltre, questo deve essere condiviso per tutta la sua estensione e non solamente in parte. Due virtual face possono, tuttavia, intersecarsi in maniera irregolare. In questa fase dell'algoritmo tali anomalie vengono investigate e opportunamente gestite; esistono due tipi di intersezioni illegali: quelle del primo tipo si presentano quando un punto interno di uno spigolo di una faccia coincide con un punto interno di uno spigolo di una seconda faccia; quelle del secondo tipo si verificano quando, in assenza di un intersezione illegale del primo tipo, un vertice di una delle due facce giace sul piano dell'altra e contemporaneamente esiste un punto interno ad entrambe. Per maggior chiarezza, in figura 2.3 viene riportata la corretta giunzione di due facce, in figura 2.4 si mostrano due casi di intersezione illegale di prima tipo ed in figura 2.5 vengono rappresentate quattro situazioni di intersezione illegale del secondo tipo. Per quanto riguarda la prima tipologia, il problema è impossibile da risolvere in questa fase dell'algoritmo a causa della mancanza di informazioni; pertanto, artificialmente, viene introdotto un nuovo spigolo (cutting edge) tale che le due facce possano essere suddivise e l'anomalia eliminata. L'elaborazione delle intersezioni illegali del primo tipo contribuisce, quindi, all'aumento del numero delle virtual face.

Le anomalie della seconda tipologia vengono trattate differentemente dalle prime. Come si può notare dalla figura 2.5, una delle due facce contiene sempre entrambi gli estremi dello spigolo "e"; per la soluzione di questa intersezione illegale tale faccia deve essere eliminata.

2.1.5 Determinazione dei 2-cycle e dei virtual block

Durante questo passo dell'algoritmo viene sostanzialmente ripetuto ciò che è stato precedentemente eseguito al terzo passo. La differenza sostanziale risiede nell'aumento della dimensione geometrica del problema; nel caso già affrontato venivano eseguite operazioni su geometrie bidimensionali, in questa fase la geometria trattata è di tipo tridimensionale. Per ogni spigolo si genera una lista radialmente ordinata di virtual face su di esso incidenti. Si procede quindi percorrendo le virtual face analogamente a quanto avveniva al punto tre per gli spigoli, generando quindi dei cicli che, invece di delimitare superfici, delimitano adesso regioni solide. Ciascun ciclo di facce divide lo spazio in due regioni, la scelta della regione legata al ciclo viene effettuata, analogamente a quanto fatto in precedenza per i cicli di spigoli, in maniera convenzionale. Altre analogie legano

questo passaggio al numero tre: la ricerca di cicli a termine allorché tutte le facce risultano percorse su entrambi i lati; la stessa faccia percorsa su entrambi i lati all'interno del medesimo ciclo prende il nome di bridge e viene eliminata, provocando analoghe conseguenze al caso precedente; i cicli, una volta determinati, vengono archiviati nell'insieme dei virtual block.

2.1.6 Costruzione delle soluzioni

Utilizzando un albero decisionale, i virtual block vengono catalogati secondo regole elementari. Essi possono assumere lo stato di solido o di foro; assegnando tale stato in modo ragionato è possibile costruire tutte le soluzioni ricollegabili al wire frame di partenza. Per maggiori dettagli su questo passaggio finale, si rimanda il lettore direttamente a [11].

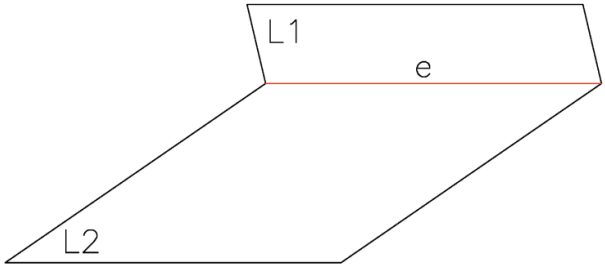


Figura 2.3 – Due virtual face correttamente giuntate

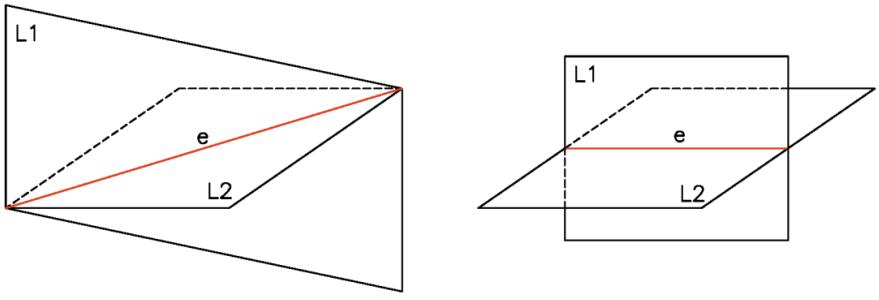


Figura 2.4 – Intersezioni irregolari di primo tipo

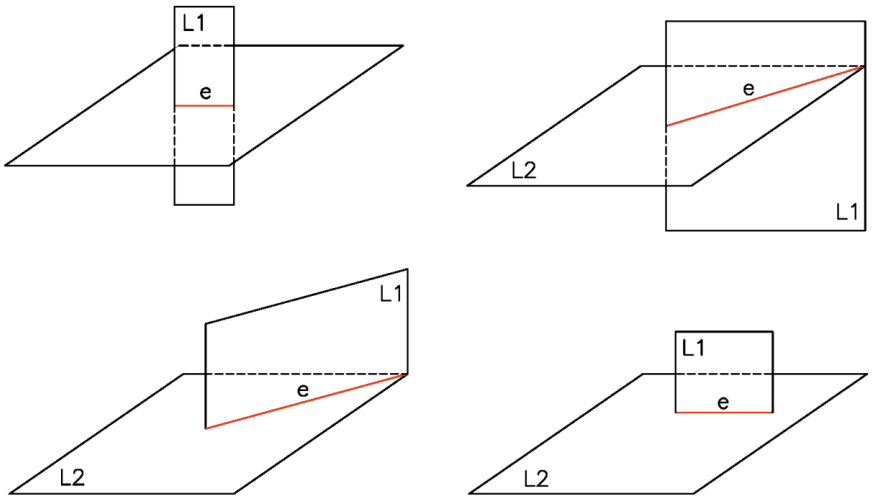


Figura 2.5 – Intersezioni irregolari di secondo tipo

2.2 Ricostruzione del modello wireframe a partire da proiezioni ortogonali

Un anno più tardi, Wesley e Markowsky pubblicano un nuovo lavoro, intitolato “Fleshing out projections” [11], che essi stessi considerano un’estensione del precedente. Lo scopo finale dei due lavori è quello di fornire un metodo per la ricostruzione di geometrie tridimensionali di oggetti poliedrici partendo da un certo insieme di proiezioni bidimensionali ad essi riferite. Le viste bidimensionali di partenza, ipotizzate da Wesley e Markowsky, sono per lo più rappresentate da disegni tecnici. É importante notare come, nelle tavole tecniche, oltre alle informazioni sotto forma di geometrie planari, si trovano molti altri dati potenzialmente utili alla ricostruzione. Le quotature, le annotazioni e le viste in sezione sono soltanto alcune di esse, senza contare che la capacità di “leggere” nel disegno le varie tipologie di linea (continue, tratteggiate, etc.) costituisce un grande vantaggio per la corretta interpretazione del disegno stesso.

Quanto appena affermato è fortemente condizionato sia dalla convenzione seguita nella stesura della tavola, sia nella fedeltà con cui sono state applicate le norme che disciplinano tale lavoro. Per tali motivi, gli autori hanno deciso di fornire un modello indipendente da tali vincoli. La conseguente assenza di parte delle informazioni rende il modello più complesso anche se di più generale impiego. L’impatto maggiore di questa scelta si nota nella fase di sintesi dei

risultati, questo accade poiché le lacune lasciate dai vuoti informativi introducono ambiguità durante la loro generazione. L'insieme delle soluzioni ricavate da un'attenta analisi di tutti i dati presenti nelle viste sarà pertanto un sottoinsieme delle soluzioni generate dal presente algoritmo. In particolare la presenza di molti assi e piani di simmetria provoca una crescita notevole delle geometrie solide elaborate. Ad esempio, un caso di moltiplicazione delle soluzioni può essere verificato in figura 2.6, dove ne vengono mostrate le proiezioni ortogonali ed il relativo pseudo-wire frame; in figura 2.7 ne vengono invece mostrate alcune soluzioni. I casi di reale interesse (figura 2.8) sono tuttavia ben diversi dagli esempi mostrati nelle figure 2.6 e 2.7, questo fa si che nella pratica quotidiana non vi sia il costante pericolo di esplosioni combinatorie tali da rendere inservibile il modello di ricostruzione.

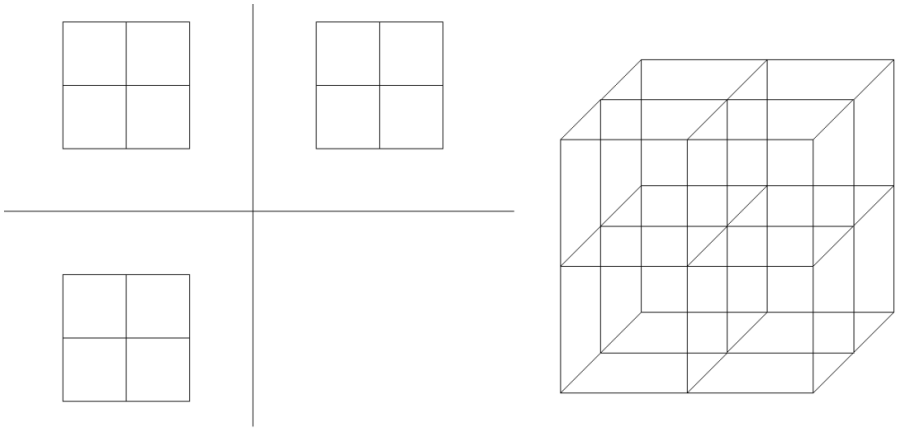


Figura 2.6 – Proiezioni e wire frame di un oggetto contenente piani di simmetria.

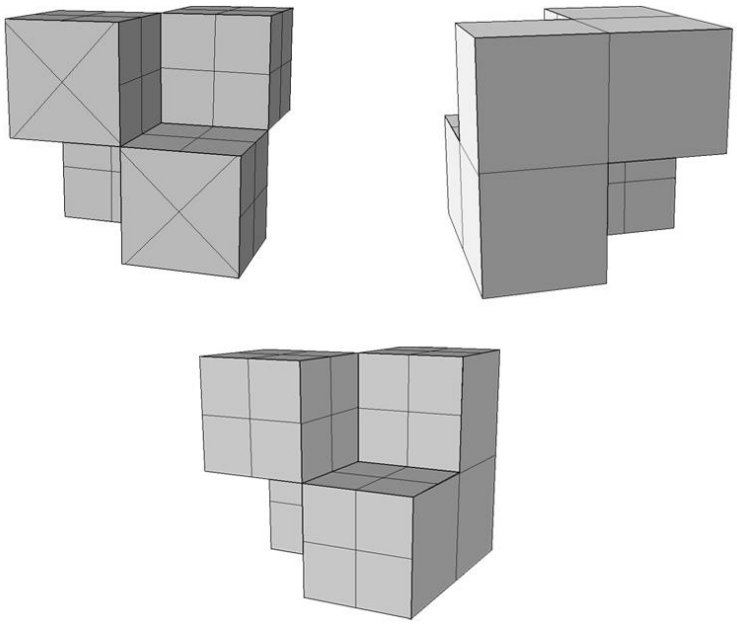


Figura 2.7 – Soluzioni legate alle proiezioni di figura 2.6

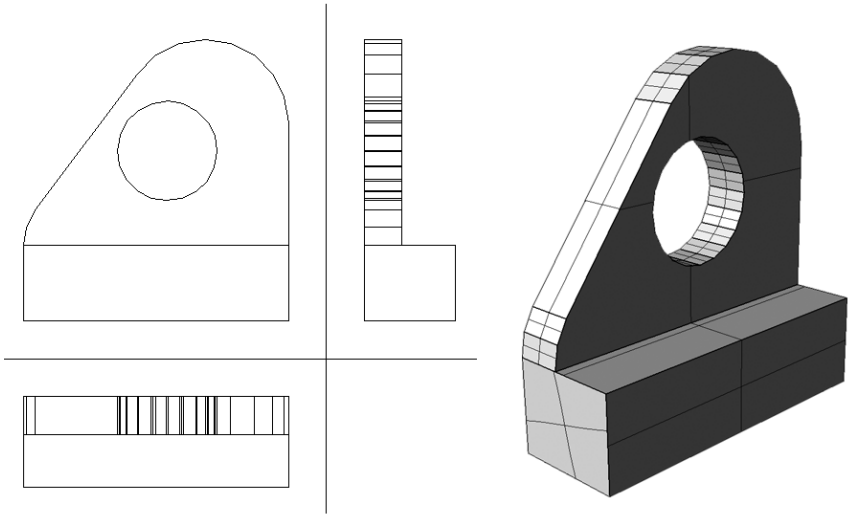


Figura 2.8 – Particolare privo di piani di simmetria

Come per il precedente paragrafo, verranno di seguito riportate le fasi salienti dell'algoritmo, rimandando, per maggiori approfondimenti, al terzo capitolo dove i concetti più importanti verranno ripresi ed esaminati più nel dettaglio. Per facilitare ulteriormente la comprensione, anche in questo caso, in appendice "A" vengono riportati i concetti topologici e geometrici che sono alla base del presente lavoro.

Le prime fasi dell'algoritmo sono relative alla conversione di un insieme di proiezioni in un modello tridimensionale, il risultato di questa trasformazione non è altro che un modello wire frame che rappresenta un sovrainsieme di quello appartenente all'oggetto della ricostruzione. Per tale motivo questo primo modello viene etichettato come pseudo wire frame. La costruzione del modello pseudo wire-frame, preceduta ancora una volta da una fase di controllo dei dati di input, è descritta nei paragrafi seguenti.

2.2.1 Controllo dei dati di input

La base dati dell'algoritmo deve essere costituita da un insieme di viste bidimensionali scelte in modo tale che rappresentino l'intero oggetto da ricostruire. Ciascuna vista viene archiviata come l'insieme dei suoi nodi (vertici) e dei suoi archi (spigoli) nonché con una matrice di trasformazione in grado di posizionare, a meno della terza coordinata, tale vista nello spazio. Per quanto concerne l'introduzione di eventuali controlli, la linea guida tracciata nel precedente paragrafo si mantiene valida anche per questo modello. Tuttavia è necessario ricordare che la natura combinatoria dell'algoritmo di gestione delle proiezioni è molto sensibile alla quantità di informazioni fornite, pertanto è fortemente consigliata l'eliminazione di tutti quei dati che generano una ridondanza di informazioni.

2.2.2 Costruzione di uno pseudo scheletro tridimensionale

Nel corso di questa fase viene ricavato un insieme di vertici dato dall'unione di quelli di classe I e da una parte di quelli di classe II; questo avviene mediante la proiezione dei vertici di ciascuna vista perpendicolarmente alla vista stessa. La classe I viene interamente rilevata ed identifica i vertici frutto dell'intersezione di spigoli non complanari; la classe II, destinata ai restanti complanari, non può essere completata in questo passaggio. Ad ogni modo è ancora troppo presto per etichettare i vertici come appartenenti all'una o all'altra classe, l'algoritmo si limita quindi ad archiviare l'insieme generale. Tale insieme,

completato con gli elementi mancanti di classe II (determinati nel prossimo passaggio) prende il nome di insieme dei candidate vertex. Il termine "candidate" indica che soltanto alcuni di essi saranno in grado di superare tutti gli step algoritmici per andare, infine, a costituire l'insieme dei vertici dell'oggetto ricostruito.

2.2.3 Costruzione di uno pseudo wire frame

Lo scheletro di vertici è quindi pronto ad accogliere gli ultimi candidate vertex e, soprattutto, gli spigoli (edge) che andranno a completare la struttura dello pseudo wire frame. Gli spigoli tridimensionali vengono introdotti sulla base delle informazioni presenti nelle proiezioni di partenza; l'eventuale intersezione di tali spigoli nello spazio viene calcolata ed archiviata come candidate vertex andando a completarne l'insieme. Analogamente a quest'ultimo, l'insieme degli spigoli generati prende il nome di insieme dei candidate edges. Abbinando le informazioni sui candidate vertex a quelle sui candidate edge si ottiene la struttura dello pseudo wire frame. La memoria del modello deve, a questo punto, contenere una corrispondenza tra gli elementi delle viste bidimensionali e quelli dello pseudo wire frame.

2.3 Sviluppi più recenti

Molti degli studi successivi a quelli appena presentati fanno riferimento a tali lavori. Gli sviluppi più importanti apportati negli anni successivi hanno seguito la linea tracciata da Wesley e Markowsky, incentrandosi particolarmente sulla gestione di forme più complesse. Tuttavia, dato l'obiettivo primario di sviluppare tecnologie applicabili nel settore della progettazione meccanica, lo sviluppo verso geometrie di grado superiore al primo si è arrestato al secondo ordine (coniche). Di seguito si descrivono due lavori, proposti nell'ordine cronologico con cui sono stati pubblicati, che rappresentano l'evoluzione negli ultimi dieci anni. Le tecniche di ricostruzione descritte nel secondo lavoro possono quindi essere considerate una delle più recenti espressioni dello stato dell'arte sulla ricostruzione da proiezioni ortogonali esatte. Come risulterà dai successivi paragrafi, è interessante come nessuna delle tecniche menzionate sia in grado di gestire un dato iniziale di tipo "inesatto".

2.3.1 Gestione delle curve coniche

Nel 2001, Liu et al. [19] producono un lavoro in grado di gestire la ricostruzione a partire da coniche il cui asse non deve necessariamente essere parallelo o ortogonale ai tre piani coordinati ma deve soltanto giacere su almeno uno di essi. Il metodo proposto richiede inoltre che i punti estremi dell'asse siano definiti e di coordinate note in ciascuna delle proiezioni di partenza. L'identificazione di tali "endpoint" avviene per mezzo di operazioni di preprocessing sul dato vettoriale in ingresso; questa operazione comporta delle complicazioni in quanto il dato in ingresso è vincolato ad avere una forma rigorosa, spesso e volentieri non riscontrabile nei disegni realizzati nella pratica comune. Inoltre il metodo applica una tecnica di riconoscimento delle coniche tra le varie viste basata sul confronto degli assi di inerzia; risultano pertanto non ricostruibili le coniche a base ellittica poiché i loro assi di inerzia sono diversi per definizione.

Dal punto di vista procedurale l'algoritmo sviluppato da Liu et al. si basa sui seguenti punti principali:

1. pre-processamento dei dati vettoriali 2D;
2. generazione di un modello wireframe 3D;
3. generazione di un set di facce compatibili con il modello di cui al punto 2;
4. generazione delle possibili soluzioni "solide" ottenibili mediante combinazione delle facce determinate al punto 3.

Come si nota, lo schema concettuale è praticamente identico a quello proposto da Wesley e Markowsky, andandosi a differenziare prevalentemente nel secondo punto con l'introduzione di un metodo che gli autori chiamano "conjugate diameter method". Il metodo prevede l'identificazione delle tre proiezioni della conica nelle tre viste mediante una tecnica di matching riconducibile alla comparazione delle proiezioni del suo bounding box 3D e la successiva ricostruzione per mezzo di considerazioni di natura geometrica sulle caratteristiche inerziali delle curve.

Per quanto riguarda il dato in ingresso, la metodologia proposta richiede quindi dati di tipo vettoriale, a tal proposito gli autori specificano esplicitamente che la ricostruzione a partire da dato raster 2D è possibile, ma soltanto dopo aver estrapolato e riordinato le informazioni che devono essere di tipo vettoriale.

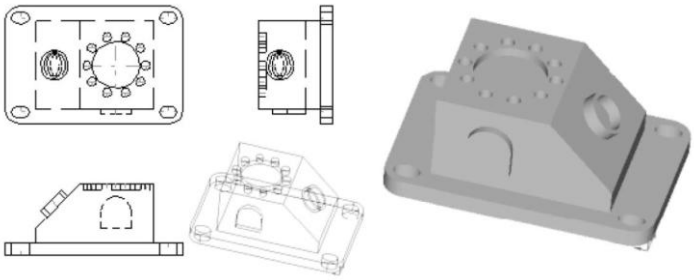


Figura 2.9 – Particolare ricostruito con la tecnica elaborata da Liu et al.

2.3.2 Ricostruzione con approcci di tipo “human-like”

Cinque anni più tardi, nel 2006, Gong et al. presentano un approccio che intende affrontare il problema tentando di emulare la logica con cui un operatore procederebbe manualmente alla ricostruzione [28]. Pur rimanendo nel dominio delle soluzioni fornite dalla metodologia precedentemente descritta, Gong introduce le curve di Bezier quadratiche nel tentativo di semplificare il processo di generazione delle entità tridimensionali. Tuttavia il metodo proposto presenta delle limitazioni dovute alla difficoltà di estrarre un dato coerente in termini di corrispondenze geometriche tra le rappresentazioni della stessa entità nelle tre viste a disposizione. Proprio a causa di queste difficoltà, nel caso in cui l’elemento da ricostruire sia costituito da una curva chiusa, mancante di due endpoint, la tecnica presentata da Gong et al. non riesce a fornire una soluzione.

In altre parole, per applicare l’algoritmo proposto da Gong, occorre che ciascuna curva sia definita da due endpoint; nel caso di curve chiuse sarà quindi necessario suddividere le stesse in coppie di curve aperte avendo cura che i tratti di curva generati dalla suddivisione abbiano tutte le rispettive corrispondenze nelle tre viste.

Ciò nonostante la tecnica proposta da Gong risulta più lineare e di più semplice applicazione rispetto a quella proposta da Liu aprendo anche uno spiraglio alla gestione delle curve di ordine superiore (un tipo esempio caratteristico della progettazione meccanica è rappresentato dal perimetro di intersezione tra due cave circolari).

Ovviamente lo schema di ricostruzione resta invariato rispetto a quelli precedentemente descritti, l’innovazione portata da Gong consiste nella tecnica con cui le entità 2D vengono tradotte nello spazio tridimensionale.

Per ciascuna entità, le cui corrispondenze tra le viste vengono stabilite per mezzo di considerazioni sui bounding box, Gong et al. procedono ad una classificazione che prevede due classi di oggetti: segmenti e curve coniche.

La classificazione e la successiva ricostruzione delle entità 3D sono legate a due concetti fondamentali ai quali Gong fa riferimento:

1. la proiezione di un qualunque tratto rettilineo comunque orientato nello spazio avrà come rappresentazione in almeno due delle tre proiezioni, un segmento, nella terza potrà degenerare in un punto;
2. la proiezione di un qualunque tratto di curva conica comunque orientata nello spazio avrà come rappresentazione in almeno una delle tre proiezioni una curva conica, nelle rimanenti due potrà degenerare in un segmento.

Con il supporto di questi due concetti, per ciascuna delle due tipologie di oggetto gli autori forniscono una classificazione nella quale la discriminante tra le possibili configurazioni è, appunto, costituita dall'orientamento dell'oggetto rispetto ai tre piani coordinati (Figura 2.10 e Figura 2.11).

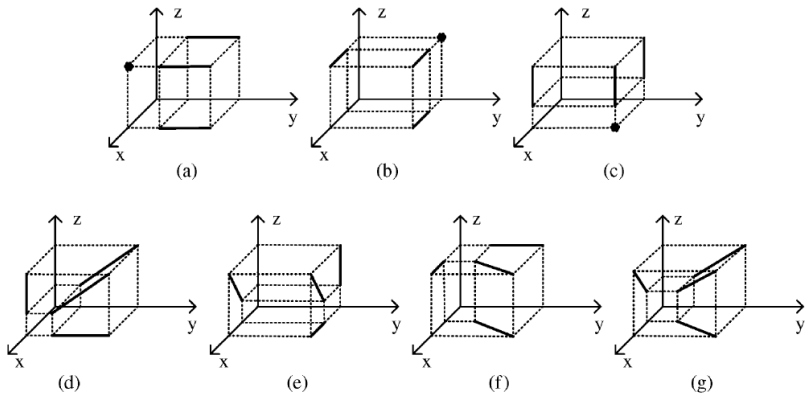


Figura 2.10 – Classe dei segmenti nel lavoro di Gong et al.

Il passo successivo dell'algoritmo proposto da Gong et al. Prevede la definizione di tre grafi planari, uno per ciascuna delle tre proiezioni ortogonali originali. Per "grafo planare" gli autori intendono la struttura topologica di ciascuna delle tre proiezioni originali in cui in nodi si identificano con i vertici e gli archi con le varie entità geometriche che compongono la vista.

I tre grafi così ricavati vengono forniti in ingresso ad un algoritmo che ne esamina i vari archi permettendo l'estrazione di tutti i set di tre proiezioni degli oggetti tridimensionali, segmenti e curve (configurazioni mostrate in Figura 2.10 e Figura 2.11), che comporranno il wireframe 3D da ricostruire.

Dal punto di vista computazionale, la ricerca delle configurazioni di cui alla Figura 2.10 e Figura 2.11 avviene per mezzo dell'esplorazione di tre alberi, uno per ciascuna vista, mediante l'algoritmo "*depth first search* (DFS)" [29].

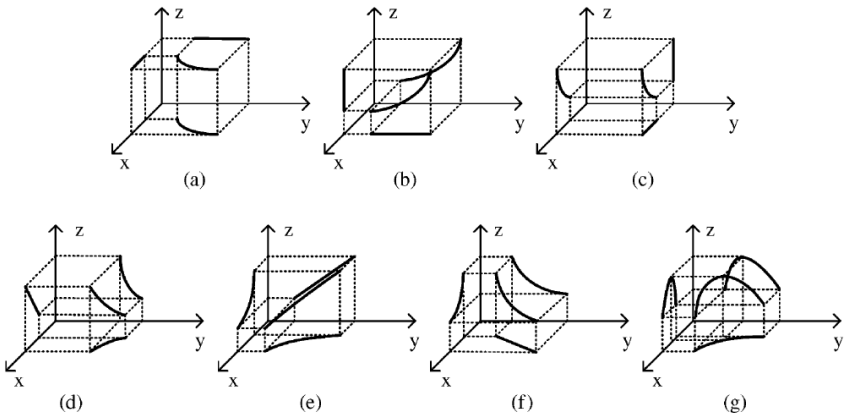


Figura 2.11 – Classe delle curve coniche nel lavoro di Gong et al.

Come si deduce dal nome stesso, il DFS effettua una ricerca in profondità nel grafo sul quale viene applicato. Partendo da un nodo del grafo, l'algoritmo propaga la sua ricerca prediligendo la percorrenza degli archi che lo conducono alla maggiore profondità di esplorazione; una volta giunto al livello più profondo l'algoritmo recupera e percorre in maniera iterativa le eventuali "strade alternative" abbandonate nei percorsi precedenti (Figura 2.12).

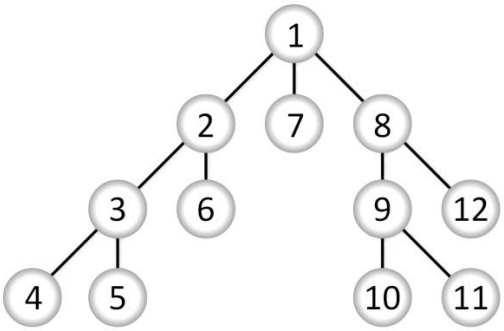


Figura 2.12 – DFS, ordine di esplorazione dei nodi

A titolo di esempio si riporta in Figura 2.13 l'albero decisionale relativo alla proiezione frontale.

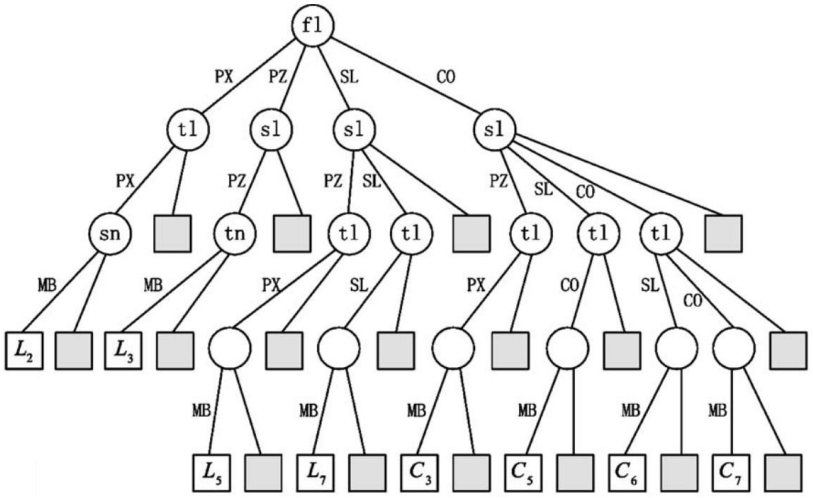


Figura 2.13 – Albero delle configurazioni per la vista frontale

In Figura 2.13 si impiega la simbologia di seguito descritta:

- fl, insieme degli archi del grafo relativo alla proiezione orizzontale;
- sl, insieme degli archi del grafo relativo alla proiezione laterale;
- tl, insieme degli archi del grafo relativo alla proiezione frontale;
- fn, insieme dei nodi del grafo relativo alla proiezione orizzontale;
- sn, insieme dei nodi del grafo relativo alla proiezione laterale;

- tn, insieme dei nodi del grafo relativo alla proiezione frontale;
- PX, oggetto di tipo “segmento” parallelo all’asse X;
- PY, oggetto di tipo “segmento” parallelo all’asse Y;
- PZ, oggetto di tipo “segmento” parallelo all’asse Z;
- SL, oggetto di tipo “segmento” sghembo rispetto ai tre assi;
- CO, oggetto di tipo “curva”;
- MB, corrispondenza individuata.

Nelle “foglie” dell’albero di Figura 2.13 si notano invece simboli di tipo L rappresentanti i link lineari 3D mostrati in Figura 2.10 e simboli di tipo C che identificano le curve 3D elencate in Figura 2.11.

Ad esempio, l’albero di Figura 2.13 può essere utilizzato per determinare un link lineare di tipo L_2 (configurazione b di Figura 2.10) se,

1. partendo da un oggetto di tipo “segmento” contenuto nel piano orizzontale e parallelo all’asse X,
2. se ne identifica la corrispondenza sul piano frontale in un secondo segmento parallelo anch’esso all’asse X ed infine
3. si individua una loro corrispondenza in un nodo sul piano laterale;

tale procedura, quindi, permette di definire una corrispondenza che porta alla generazione di un oggetto “segmento” di tipo L_2. In Figura 2.14 si riporta una rappresentazione grafica del procedimento sopra descritto a titolo di esempio; la freccia rossa identifica il passaggio dal primo al secondo punto della procedura, la verde mostra il raggiungimento del terzo punto ed infine la blu segnala la corretta identificazione delle corrispondenze tra le tre viste e definisce il tipo di oggetto che si è individuato.

La prosecuzione dell’algoritmo di Gong ha a che vedere con la determinazione delle caratteristiche geometriche degli oggetti tridimensionali individuati: mentre per i segmenti rettilinei è sufficiente conoscere i punti terminali delle proiezioni sui tre piani per poter ricostruire l’oggetto, nel caso delle curve ciò non corrisponde a verità poiché sono necessarie ulteriori informazioni per la ricostruzione della geometria corretta della curva nello spazio 3D.

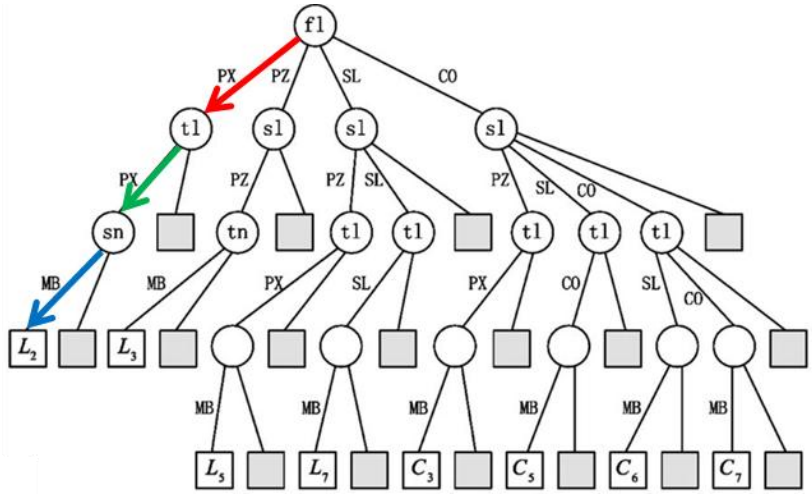


Figura 2.14 – Esplorazione dell'albero delle configurazioni

A tale proposito, nel loro lavoro, gli autori fanno ricorso alle curve quadratiche nella rappresentazione di Bezier. Sfruttando la proprietà di invarianza delle curve di Bezier nelle trasformazioni affini [30] gli autori ricostruiscono le curve 3D partendo dalle loro rappresentazioni nelle tre proiezioni di partenza. Come ultimo passaggio della loro procedura, gli autori propongono l'utilizzo di un metodo [20] per la rimozione degli elementi patologici generati.

Come affermato in fase di presentazione di questi due lavori, pur trattandosi di studi recenti, nessuno di essi risulta essere capace di gestire le due criticità maggiori che il presente lavoro si propone di affrontare:

1. dati in ingresso di tipo raster e privi di corrispondenze;
2. ricostruzione di geometrie tridimensionali di ordine superiore al secondo.

2.4 Ricostruzione mediante l'uso di template

Nel 2006 Kara ed Eramo [31] propongono una metodologia (Figura 2.15) di ricostruzione basata sull'impiego di modelli standard iniziali. La loro procedura di ricostruzione si basa, infatti, sulla combinazione di schizzi a mano libera prodotti dal designer con modelli tridimensionali predefiniti che costituiscono veri e propri "template".

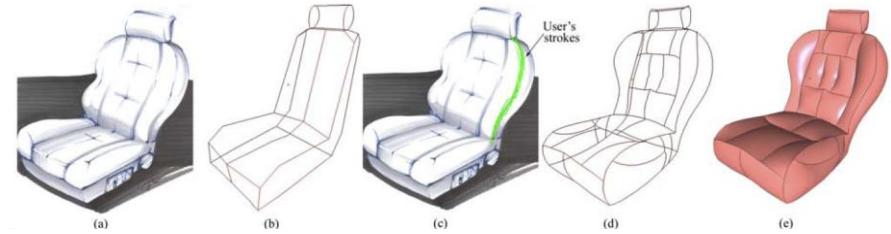


Figura 2.15 – Flusso del lavoro nella metodologia di Kara ed Eramo

L'intento è quello di fornire al designer uno strumento capace di produrre modelli computazionali 3D limitando al minimo l'invasività della tecnica di ricostruzione nella fase in cui il designer produce i disegni di stile cartacei. La fase di disegno concettuale su carta risulta quindi priva di sostanziali alterazioni. Al termine di questa fase i disegni prodotti dal designer vengono scannerizzati e trasformati in una serie di immagini bidimensionali.

La prima fase del vero e proprio processo di ricostruzione prevede l'allineamento delle immagini prodotte dalla scansione dei disegni realizzati a mano con il template di riferimento (Figura 2.16).

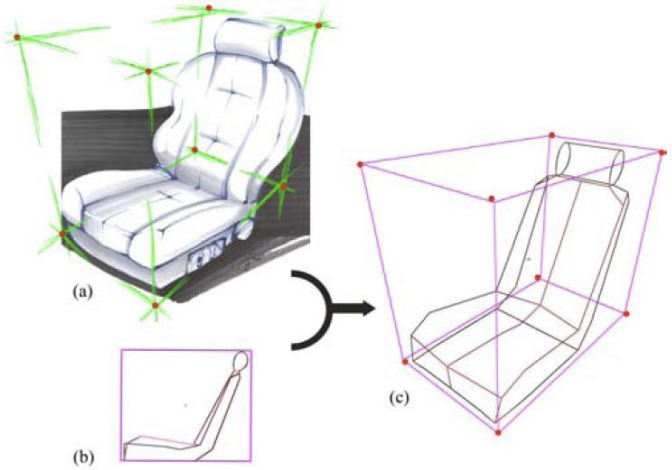


Figura 2.16 – Allineamento di template e immagine

Successivamente, mediante l'impiego di strumenti di input digitale (quali mouse o tavoletta grafica) le varie curve dei disegni scansionati debbono essere "ricalcate" (Figura 2.15c) in modo da permettere alle curve costituenti il template di modificarsi adeguandosi alla geometria rappresentata dal designer sul supporto cartaceo.

E' importante sottolineare come in questa fase possono essere utilizzate più immagini provenienti da diversi disegni cartacei ad esempio allo scopo di definire in modo più esatto la geometria di alcuni particolari/zone del modello tridimensionale.

Una volta definite tutte le curve del modello, si procede alla rigenerazione delle patch di superfice del modello stesso, il risultato a questo punto sarà un modello a superfici di primo tentativo (Figura 2.17) sul quale occorrerà lavorare al fine di produrre un risultato che interpreti nel modo migliore possibile il set di disegni bidimensionali iniziale (Figura 2.18).

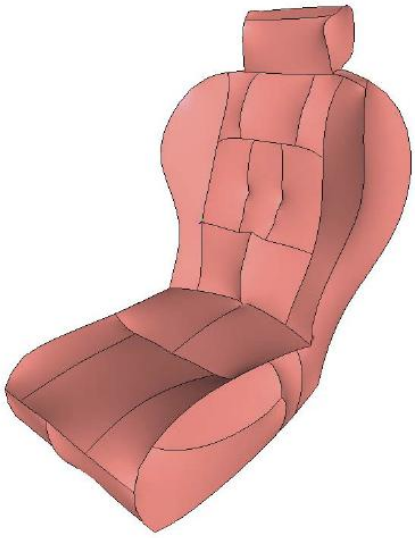


Figura 2.17 – Patch di primo tentativo

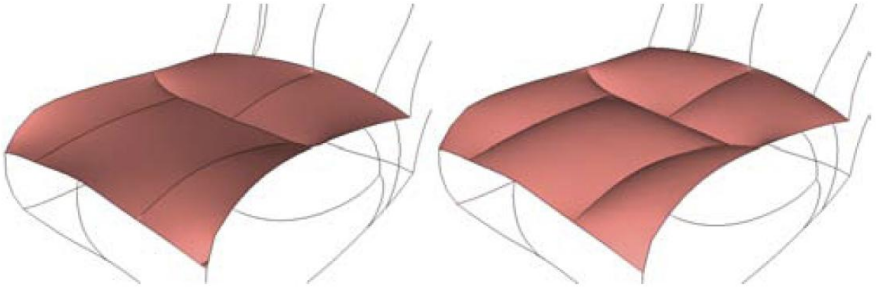


Figura 2.18 – Modellazione delle patch in post-produzione

Nel loro lavoro Kara ed Eramo propongo un set di strumenti piuttosto limitati per la modifica del modello a superfici sopra citato; tuttavia questo risulta abbastanza secondario rispetto al loro obiettivo. Il fulcro del loro lavoro risulta infatti essere l'idea di modellare un template al fine di ottenere velocemente una rappresentazione tridimensionale corretta di un disegno di stile bidimensionale.

Dal lato operativo, una delle fasi più interessanti è senza dubbio quella della procedura di allineamento dell'immagine bidimensionale con il template tridimensionale.

Dal momento che il disegno di stile può essere generato assumendo un punto di vista generico gli autori forniscono una procedura capace di risalire a tale punto di vista. Interpretando il disegno iniziale è possibile risalire alla posizione, all'orientamento ed agli altri parametri caratteristici della camera "virtuale" che genererebbe l'immagine rappresentata dal designer nello schizzo.

Per determinare tutte queste informazioni gli autori introducono una fase interattiva in cui viene chiesto all'utente di disegnare il bounding box "virtuale" dell'oggetto rappresentato nell'immagine scansionata; la conoscenza dei vertici del bounding box permette di risalire ai parametri caratteristici della camera.

In particolare gli autori utilizzano il metodo proposto da Forsyth e Ponce [32] che permette di mappare le coordinate omogenee spaziali di un punto $P = [x\ y\ z\ 1]^T$ sulle coordinate omogenee del piano immagine $p = [u\ v\ 1]^T$ nel seguente modo:

$$p = \frac{1}{s} K[R\ t]P$$

Dove s rappresenta un fattore di scala, R e t sono i ben noti parametri estrinseci della camera, mentre K rappresenta la matrice dei parametri intrinseci:

$$K = \begin{bmatrix} \alpha & -\alpha\cot(\theta) & u_0 \\ 0 & \dfrac{\beta}{\sin(\theta)} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

con u_0 e v_0 coordinate del centro della camera, α e β fattori scala rispettivamente per gli assi u e v e θ angolo compreso tra u e v (*skew angle*).
Forsyth e Ponce procedono al calcolo dei parametri estrinseci ed intrinseci in due passaggi successivi, in prima istanza calcolano una matrice di proiezione M (dimensione $3x4$):

$$M = \frac{1}{s} K[R\ t]$$

E successivamente procedono alla stima dei parametri intrinsici ed estrinseci a partire da M.

Per la prima parte, ovvero per il calcolo della matrice M utilizzano il metodo di Abdel-Aziz e Karara [33] che risolve un sistema dove n punti nello spazio tridimensionale vengono associati ad n punti sul piano immagine formando un sistema di 2n equazioni omogenee in 12 variabili. Ovviamente se il numero di punti è uguale o superiore a 6 il sistema risulta determinato. Nel caso di Kara ed Eremo non si pone nessun problema poichè i punti in questione sono gli otto vertici del bounding box.

Il secondo passaggio permette l'estrazione di s, K, R e t dalla matrice M con la conseguente determinazione del punto di vista al quale si desiderava risalire.

Ovviamente ogni qualvolta si decida di utilizzare un nuovo schizzo per "modellare" il template risulta necessario ripetere la procura di allineamento dello stesso al template.

In Figura 2.19 si mostra un processo di ricostruzione della livrea di una automobile condotto tramite la metodologia presentata da Kara e Eramo. Nel loro lavoro gli autori sottolineano come la loro metodologia avrebbe permesso un abbattimento dei tempi pari a circa il 60% di quanto necessario mediante le tecniche "tradizionali".

Il risultato del loro lavoro è pertanto molto soddisfacente; tuttavia la rigidità legata all'utilizzo di template appare un vincolo troppo forte rispetto alla creatività del designer. Infatti, l'adozione di template per la ricostruzione tridimensionale limita decisamente la possibilità di spaziare liberamente alla ricerca delle soluzioni più disparate; per rendere applicabile il processo di Kara ed Eramo è infatti di vitale importanza che lo schizzo su carta non diverga in maniera importante dalla geometria del template.

L'approccio presentato all'interno del capitolo 3 risulta essere profondamente diverso in quanto permette una più ampia libertà da parte del designer essendo privo di alcun vincolo sul tipo di geometria ricostruibile.

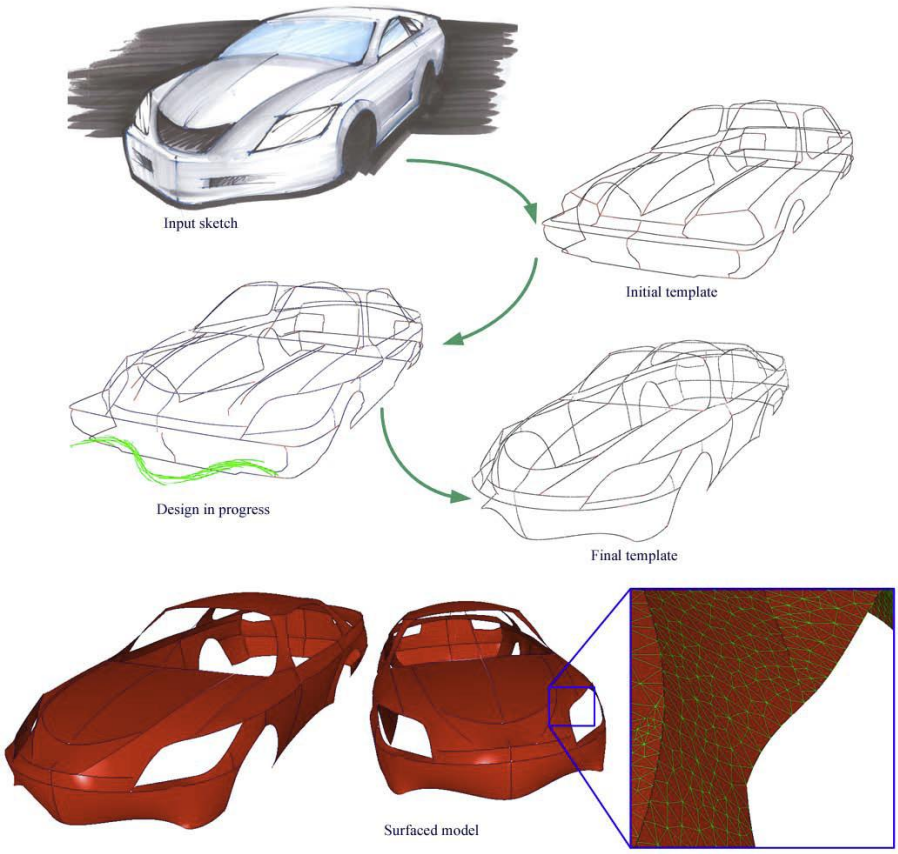


Figura 2.19 – Applicazione della metodologia di Kara ed Eramo al design in campo automobilistico

3 Metodologia

In questo capitolo viene descritta in ogni sua fase la metodologia del lavoro al centro di questo elaborato; nel corso della descrizione delle varie fasi verranno proposte, ove possibile, tecniche e metodi alternativi originati nel corso del presente lavoro.

3.1 Costruzione del modello raster 3D

La prima fase della metodologia qui presentata ha come dato di ingresso un set di disegni cartacei originali. Di fronte a questa situazione, e con la necessità di fornire come risultato finale un modello vettoriale tridimensionale, si possono individuare due diversi approcci possibili:

1. tradurre le entità bidimensionali da raster a vettoriali e combinarle successivamente per ottenere informazioni di tipo tridimensionale;
2. tradurre le entità bidimensionali raster nello spazio 3D e, successivamente, interpretare l'informazione spaziale raster e convertirla in dati vettoriali di tipo tridimensionale.

Nel corso del presente lavoro sono state esaminate entrambe le possibilità, tuttavia la seconda si è rivelata essere la migliore. Difatti, lavorare sulla traduzione di entità vettoriali 2D in geometrie 3D porterebbe con se tutta una serie di problematiche legate alla mancanza di corrispondenze tra le curve vettorializzate dalle viste di partenza. Le peculiarità della seconda soluzione sono invece da individuare prevalentemente nei vantaggi riscontrati nella traduzione del modello raster 2D in un analogo modello a tre dimensioni. La gestione delle viste di partenza con la teoria delle matrici permette, infatti, di definire una procedura di traduzione molto robusta.

Si consideri, quindi, l'immagine di partenza come risultato di una scansione in toni di grigio di un foglio di carta recante le tre viste originali disegnate a mano dal designer (Figura 3.1).

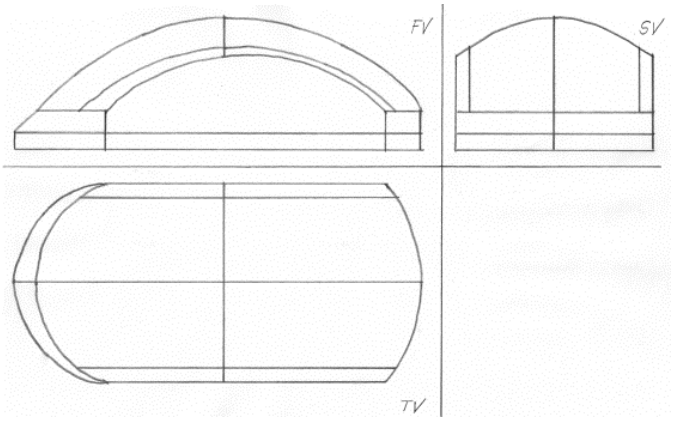


Figura 3.1 – Bozzetto di stile disegnato su supporto cartaceo

L'acquisizione deve avere una qualità sufficiente a mantenere intatta l'informazione contenuta nel disegno cartaceo; a tale scopo deve essere scelta una risoluzione di scansione tale da mantenere un buon livello di dettaglio senza compromettere la snellezza computazionale dell'algoritmo.

3.1.1 Rimozione degli errori di orientazione causati dalla scansione

Il passaggio successivo prevede la correzione dell'errore di orientazione del disegno, o di singole viste, causato da fattori quali errori dovuti alla procedura di scansione o possibili disallineamenti originariamente presenti sul disegno, qualora questo sia fatto a mano. Tale operazione viene condotta mediante individuazione del punto di origine degli assi e successiva rotazione del disegno attorno ad esso. In questo modo è possibile allineare l'estrazione delle tre viste mediante l'individuazione delle linee di demarcazione tra esse. Tali linee sono generalmente le più lunghe presenti nel disegno e pertanto possono essere identificate mediante applicazione della trasformata di Hough all'immagine acquisita (e sogliata). Come noto, infatti, la trasformata di Hough è un operatore che permette la trasformazione dal piano dell'immagine (su cui la forma è rappresentata) ad uno spazio parametrico in cui ciascun punto P corrisponde alla curva formata dai punti immagine appartenenti a tutte le possibili rette passanti per P. Se nell'immagine sono presenti più punti, approssimativamente allineati su di una retta, questa è individuata nello spazio parametrico come il punto

intersezione di tutte le curve che corrispondono alle trasformazioni dei vari punti del segmento. Maggiore è il numero di punti che concorrono a formare un segmento di retta e maggiore è il "punteggio" assegnato dalla trasformata nello spazio dei parametri. Gli assi coordinati, pertanto, sono individuabili come quelli che hanno un miglior punteggio rispetto a qualunque altra entità presente nel disegno.

Il valore del parametro di soglia viene scelto empiricamente in modo da garantire qualitativamente la massima fedeltà dell'immagine binaria al disegno originale.

La procedura di orientamento dell'immagine si compone quindi di due fasi principali:

- stima del punto di origine degli assi;
- orientazione dell'immagine.

La prima di queste due fasi prevede la definizione di una tolleranza angolare legata alla qualità di realizzazione e di successiva acquisizione del disegno. Entro tale tolleranza si procede all'individuazione della linea di demarcazione orizzontale (evidenziata in colore blu nella Figura 3.2).

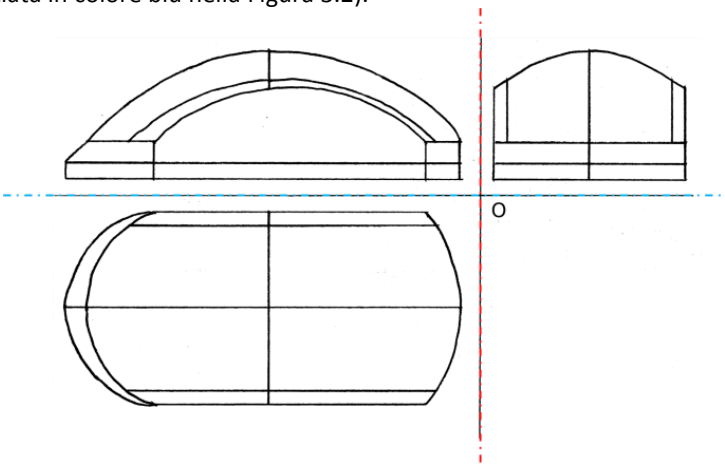


Figura 3.2 – Interpretazione del risultato della scansione

Una volta identificata la prima linea si procede con la ricerca della seconda linea di demarcazione (evidenziata in rosso nella Figura 3.2). Tale ricerca viene effettuata mediante la definizione di una ulteriore banda di tolleranza angolare come nel caso precedente. Al netto della tolleranza impostata, la ricerca avviene quindi sulle direzioni ortogonali a quella della prima linea. In questo caso, tuttavia, la banda di tolleranza risulta più ridotta in quanto l'errore da compensare

è soltanto quello dovuto alla manualità del designer; l'orientamento della scansione risulta già corretto dopo l'identificazione della prima linea.

Una volta identificata la seconda linea è possibile determinare il punto di origine attorno al quale ruotare l'immagine in modo da rendere gli assi del disegno il più possibile paralleli a quelli dell'immagine. Ovviamente, non essendo necessariamente ortogonali tra loro, non risulta generalmente possibile orientare correttamente entrambi gli assi; per questo motivo si allinea con il sistema di riferimento dell'immagine l'asse che ha ottenuto il punteggio migliore dalla trasformata di Hough.

3.1.2 Identificazione delle viste e loro estrazione

Una volta corretta l'orientazione dell'immagine, è possibile determinare le tre viste in ragione della loro posizione reciproca e della posizione del quadrante "vuoto"; tale possibilità deriva dalle convenzioni di disegno trasferite e standardizzate con il tempo nel sistema ISO.

Le tre viste vengono così estratte attraverso l'identificazione automatica del loro bounding-box ossia del rettangolo che racchiude tutti gli elementi del disegno della vista. Il calcolo dei bounding-box è possibile grazie alla gestione dell'immagine binaria in termini di componenti connessi. Le tre viste saranno infatti caratterizzate dai tre componenti connessi aventi il maggior numero di pixel e distinti dalla posizione del loro baricentro, che cadrà sempre nel quadrante di pertinenza della vista.

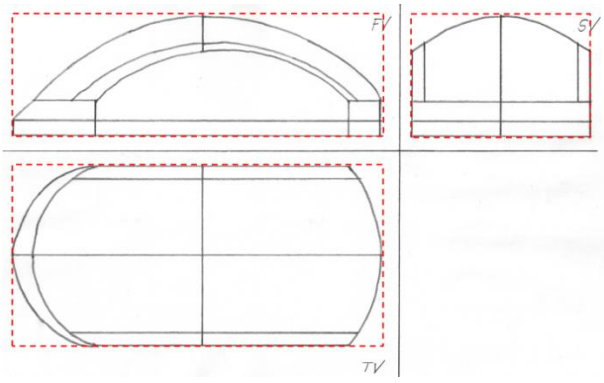


Figura 3.3 – Bounding box delle tre proiezioni

Il bounding box di ciascuna vista viene successivamente (di fatto marginalmente) corretto mediante un operazione di scalatura in modo da

uniformare le dimensioni delle tre immagini risultanti. E' infatti necessario che le dimensioni delle tre viste estratte (Top View, Front View, Side View) abbiano dimensioni compatibili per poter procedere con la loro elaborazione:

$$\{TV\}^{l\,x\,h}\,;\,\{FV\}^{l\,x\,w}\,;\,\{SV\}^{h\,x\,w} \qquad (3.1)$$

3.1.3 Operazioni di image processing sulle viste estratte

Dopo una operazione di sogliatura per riportare a livello binario le immagini scalate (l'operazione di scalatura impiega un algoritmo bicubico e quindi provoca la conversione dell'immagine da binaria a toni di grigio), si procede ad una operazione morfologica di dilatazione [34] sulle singole immagini; l'operazione viene condotta con un *kernel* di dimensioni $3x3$ avente tutti e nove i valori unitari.

Supponendo di effettuare n_s dilatazioni, i nuovi pixel di valore unitario, ottenuti all'iterazione i, risultano moltiplicati per un coefficiente d_i pari a:

$$d_i = 1 - \frac{i}{n_s + 1} \qquad (3.2)$$

E' importante notare come tutte le operazioni di dilatazione avvengono in modo iterativo, ovvero l'immagine in input al passo i consiste nell'immagine in output al passo $i - 1$. Al termine delle iterazioni vengono quindi assegnati i pesi in funzione di quanto sopra. Dal punto di vista del processamento delle immagini il peso del singolo voxel coinciderà con il suo valore di grigio (in una scala progressiva che parte da 0 per i voxel neri e termina a 1 per i voxel bianchi).

In Figura 3.4 si mostra un esempio di dilatazione avente $n_s = 3$, in particolare si evidenziano i vari step di dilatazione di una porzione di immagine, oltre a mostrare il processo nei vari passaggi reali monocromatici se ne mostra una versione a colori per favorire la comprensione (con riferimento alle immagini a colori, il rosso indica un valore del peso di 0.75, il giallo di 0.5 ed il verde di 0.25).

L'operazione di applicazione dei pesi ha lo scopo di mantenere "memoria" dei pixel bianchi nell'immagine binaria originale (ovvero dei pixel la cui appartenenza alle viste disegnate a mano dal designer è più probabile). Minore è il peso e minore la corrispondenza del pixel alle proiezioni originali; d'altro canto, l'operazione di dilatazione ha un ruolo fondamentale nella determinazione del modello wireframe 3D come sarà chiarito in seguito.

Il risultato finale di questa procedura consiste in un set di tre (una per ciascuna vista) matrici bidimensionali di dimensioni $l\,x\,h$, $l\,x\,w$ e $h\,x\,w$ rispettivamente, contenenti valori compresi tra 0 e 1.

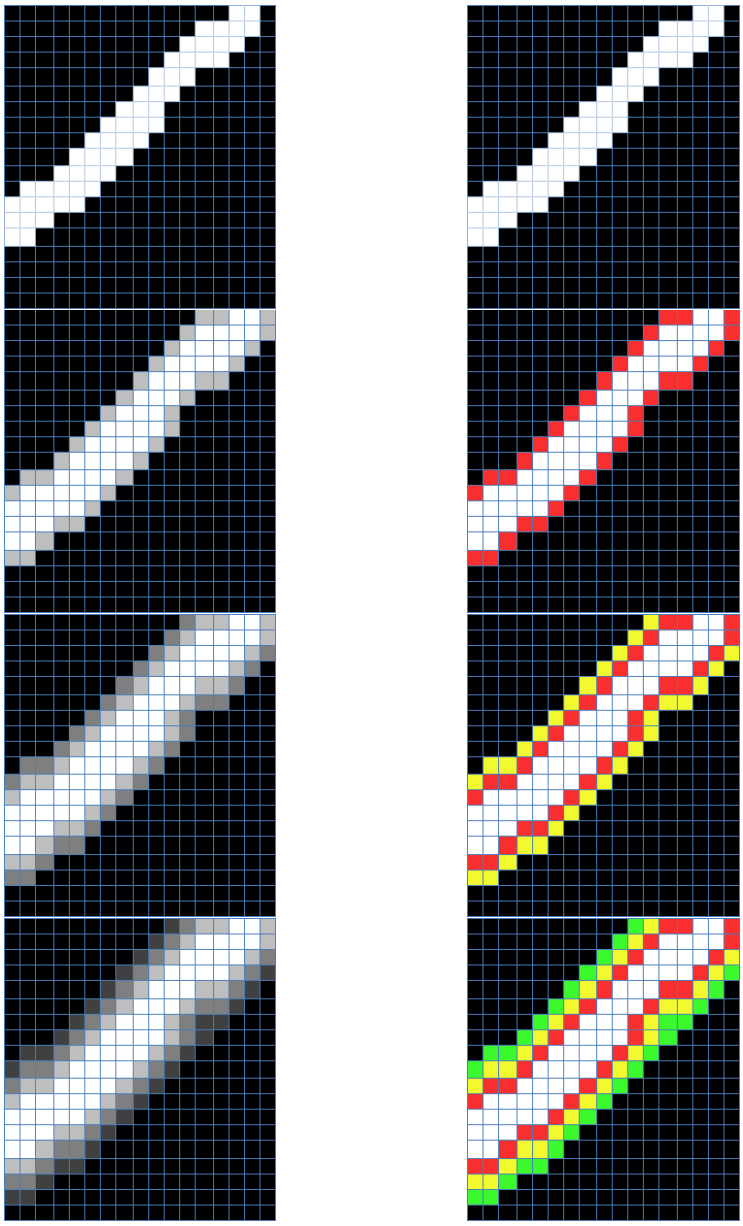


Figura 3.4 – Dettaglio dell'operazione di dilatazione ed effetto dei pesi; dall'alto in basso a sinistra si mostrano in successione: immagine originale, primo step (grigio 75%), secondo step (grigio 50%) e terzo step (grigio 25%). A destra le versioni a colori per maggior facilità di interpretazione.

3.1.4 Generazione dei modelli 3D

Le immagini ottenute, trattando come descritto in precedenza ciascuna vista nella sua forma di matrice bidimensionale, vengono quindi "estruse" lungo l'asse ortogonale al piano di giacitura della vista stessa. Le operazioni di cui al paragrafo 3.1.2 permettono la determinazione del volume che andrà ad occupare il bounding box del wireframe finale (ovvero $l\ x\ h\ x\ w$). La conoscenza delle dimensioni del bounding box finale permette, quindi, di ottenere le dimensioni caratteristiche delle tre estrusioni; il valore della lunghezza risulta infatti pari a quella, tra le tre sopra citate, che non definisce le dimensioni della relativa immagine 2D di partenza.

Prendendo ad esempio la vista TV (di dimensioni $l\ x\ h$) la lunghezza dell'estrusione sarà uguale a w.

Così facendo, si ottiene un set di tre matrici tridimensionali rappresentanti le tre immagini delle tre viste estruse lungo tutto il bounding box del wireframe 3D che si intende determinare. Ciascuna matrice sarà composta da elementi il cui valore resterà compreso tra 0 e 1.

Le tre immagini devono infine essere combinate tra loro per poter creare l'immagine finale, ovvero la rappresentazione raster del modello wireframe tridimensionale. Operativamente questo risultato viene ottenuto mediante due operazioni successive:

- normalizzazione delle tre matrici;
- moltiplicazione elemento per elemento delle tre immagini raster.

L'immagine risultante risulta essere un matrice tridimensionale i cui elementi (di fatto "voxel") avranno valore variabile tra 0 ed 1. In particolare:

- elementi con valore unitario rappresenteranno punti dell'immagine con elevata fedeltà rispetto alle viste disegnate a mano dal designer; ciò si può asserire in ragione del fatto che un valore unitario è sinonimo di corrispondenza esatta tra valori unitari appartenenti alle tre viste di partenza;
- elementi con valore nullo identificheranno il background dell'immagine;
- elementi con valore intermedio tra 0 ed 1 saranno elementi "artificiali" creati grazie alle operazioni di dilatazione delle viste originali. Essi dovranno essere utilizzati, attribuendogli un peso minore, nelle fasi successive che porteranno alla generazione del modello wireframe vettoriale vero e proprio in quanto indispensabili per una corretta rappresentazione topologica del modello wireframe raster.

In Figura 3.5 e Figura 3.6 si possono osservare due modelli wireframe raster ottenuti con la metodologia sopra descritta rispettivamente con numero di iterazioni del processo di dilatazione pari a zero e a due. E' possibile osservare come nel primo caso si abbia una perdita di informazioni che rende di fatto impossibile la procedura di ricostruzione.

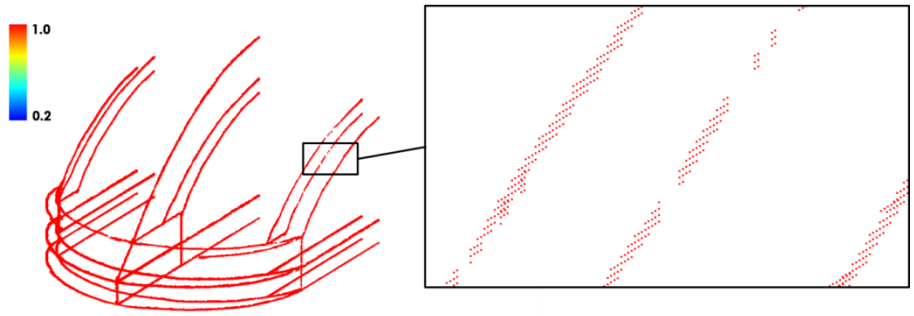


Figura 3.5 – Modello wireframe ottenuto con $N_s = 0$

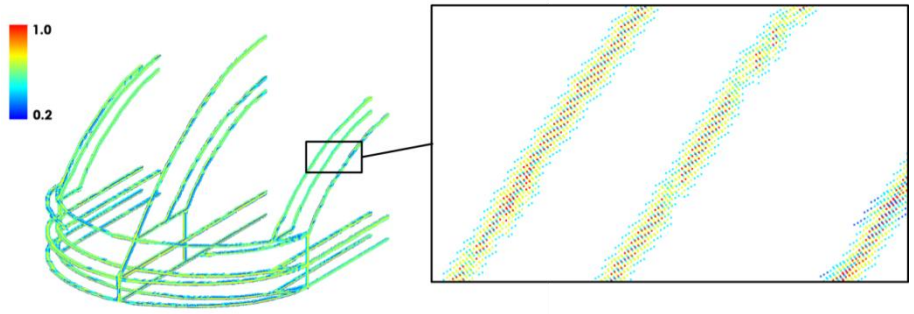


Figura 3.6 – Modello wireframe ottenuto con $N_s = 2$

3.2 Estrazione della topologia del modello raster 3D

Una volta determinata l'immagine rappresentante correttamente il modello wireframe in formato raster è possibile procedere all'analisi di tale immagine e alla determinazione della topologia dell'oggetto che detto modello rappresenta. L'estrazione di informazioni sulla corretta configurazione topologica del modello è di fondamentale importanza per la prosecuzione dell'algoritmo e per la corretta ricostruzione del modello vettoriale finale.

A tal proposito sono state elaborate e successivamente implementate e testate numerose metodologie:

1. Metodo delle distanze
2. Metodo del gradiente
3. Metodo della soglia
4. Metodo della scheletonizzazione
5. Metodo dello smoothing

Nei paragrafi successivi verranno descritti i suddetti metodi con particolare riguardo nei confronti del metodo dello smoothing rivelatosi, tra gli altri, il più efficace.

3.2.1 Metodo delle distanze

Il metodo delle distanze si basa sull'individuazione delle zone di giunzione dell'immagine 3D per mezzo della stima della distanza locale tra i voxel della nuvola ed il background. In altre parole, sia w l'indice del voxel di coordinate (x_w, y_w, z_w) appartenente al background dell'immagine, dato un voxel b di coordinate (x_b, y_b, z_b) appartenente ai voxel dell'oggetto, si calcola il valore D_b ad esso associato:

$$D_b = \min_w \sqrt{(x_b - x_w)^2 + (y_b - y_w)^2 + (z_b - z_w)^2}$$

Il risultato di questa operazione consiste nell'etichettatura di ciascun voxel mediante un valore che rappresenta la sua distanza dal voxel di background più vicino.

Con riferimento alla Figura 3.7, si mostra un esempio di applicazione del metodo; le aree della figura caratterizzate da colorazioni tendenti al rosso caratterizzano le zone di maggior spessore della nuvola. Tali zone possono essere considerate, con buona confidenza, zone di intersezione tra due, tre o più entità geometriche.

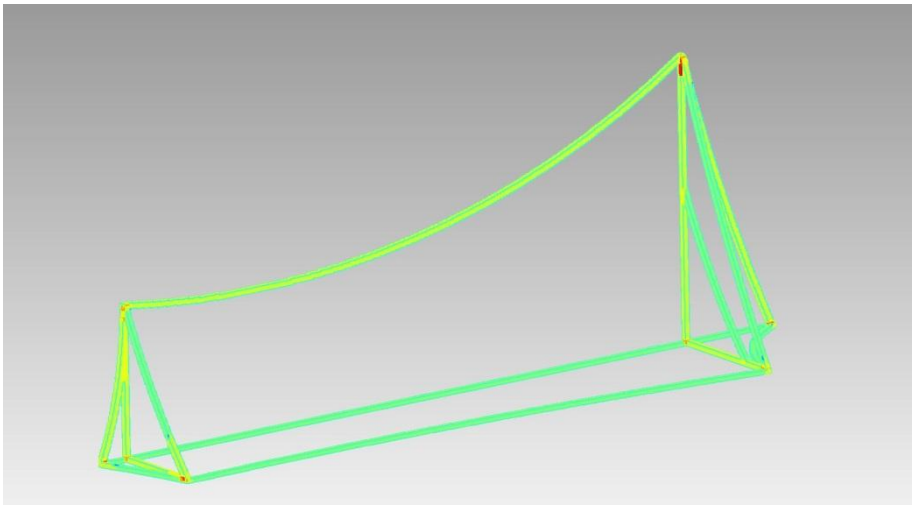


Figura 3.7 – Metodo delle distanze per il calcolo delle zone di intersezione

Il metodo si dimostra particolarmente efficacie nei casi in cui la nuvola di voxel abbia una "sezione" praticamente costante; tuttavia esso tende a fallire nei casi in cui la giunzione sia tra un numero limitato di entità (ad esempio soltanto due) come si evince dai due casi illustrati in Figura 3.8.

(a) (b)

Figura 3.8 – Metodo delle distanze – casi critici

3.2.2 Metodo del gradiente

Un metodo più performante del precedente è ottenibile mediante una successiva elaborazione dei dati ottenuti col metodo delle distanze. Questa elaborazione, detta metodo del gradiente, sfrutta infatti i valori di distanza calcolati

con il metodo sopra descritto per compiere un ulteriore passo: analizzare come variano tali valori all'interno della nuvola.

Sia Θ l'insieme dei voxel appartenenti all'oggetto; dato un voxel $b \in \Theta$ di coordinate (x_b, y_b, z_b), si calcola il valore G_b ad esso associato come segue:

$$N_{26}^{*}(b) = \{n \in \Theta \mid \max(|x_b - x_n|, |y_b - y_n|, |z_b - z_n|) = 1\}$$

$$\Delta D_h^k = D_h - D_k$$

$$G_b = \max_{n \in N_{26}} |\Delta D_b^n|$$

Dove G_b è il valore (discreto) massimo del gradiente della nuvola di voxel calcolato per ciascun punto prendendo in esame tutti i voxel appartenenti al vicinato N26 (per la definizione del vicinato N26 si rimanda al paragrafo seguente).

In Figura 3.9 si può osservare il risultato dell'applicazione del metodo del gradiente messo a confronto con l'analogo risultato ottenuto col metodo delle distanze. Le zone di colore giallo e rosso si distinguono con più nitidezza rispetto alle equivalenti zone determinate con il metodo delle distanze. Il metodo del gradiente porta, però, una criticità diversa: il presentarsi di numerosi "falsi positivi". Ad esempio, come si può notare dalla Figura 3.10, la zona inferiore, contraddistinta dal colore rosso-arancio, non risulta essere una zona di intersezione.

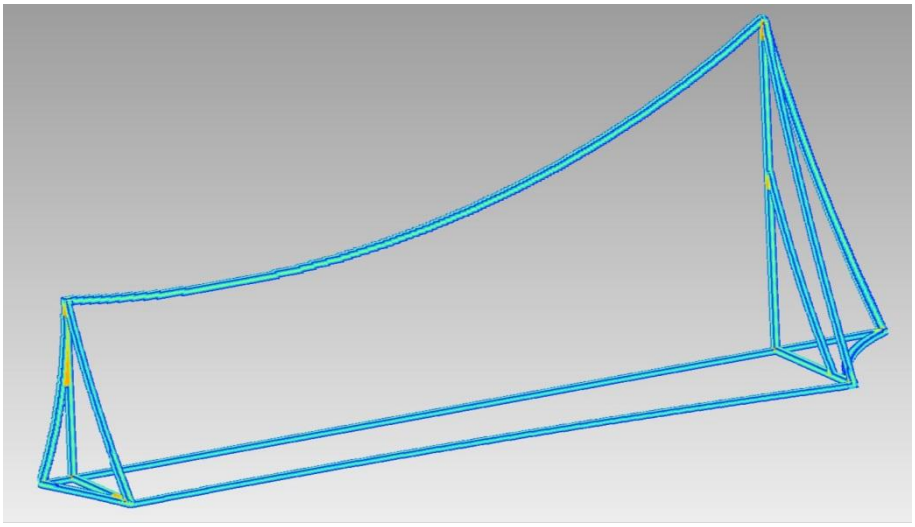


Figura 3.9 – Metodo del gradiente per il calcolo delle zone di intersezione

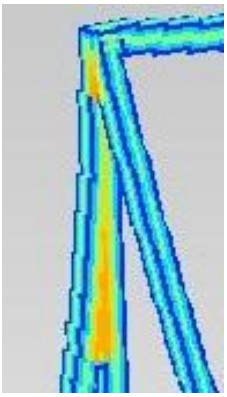


Figura 3.10 – Metodo del gradiente – casi critici

3.2.3 Metodo della soglia

Data la non completa affidabilità dei due metodi sopra descritti si è presentata la necessità di svilupparne un terzo che potesse raccoglierne i principi di base sfruttandoli in modo da produrre un risultato ancora più robusto. Per questo motivo è stato ideato e sviluppato un ulteriore metodo, definito metodo della soglia, tramite il quale massimizzare il risultato ottenibile partendo da considerazioni analoghe a quelle fatte nei precedenti paragrafi.

Rispetto ai precedenti due approcci in questa sede si è prevista la possibilità di adottare indifferentemente sia la distanza euclidea sia la distanza Manhattan [35], formalmente descrivibile come di seguito:

$$D_m(P_i, P_j) = |x_i - x_j| + |y_i - y_j| + |z_i - z_j|$$

In pratica, tale distanza si calcola come la somma delle proiezioni sui tre assi della distanza euclidea tra i due punti ai quali la si vuole applicare.

Per la comprensione di questa metodologia è inoltre utile introdurre alcuni concetti propri delle tecniche di trattamento delle immagini tridimensionali.

Un'immagine tridimensionale è definibile come una sequenza di tante immagini sovrapposte l'una con l'altra; se ipotizziamo che le immagini abbiano dimensione $n \; x \; m$ e che il numero di immagini sovrapposte sia h otteniamo una immagine tridimensionale di dimensioni $n \; x \; m \; x \; h$. In tale immagine è facile capire come un generico voxel sia circondato nello spazio 3D da una serie di altri voxel, per la precisione da ulteriori voxel che ne rappresentano una particolare definizione di vicinato [36]. In letteratura è possibile trovare una serie di questi vicinati, ognuno dei quali caratterizzato da una propria struttura e da una propria

definizione matematica [37], utilizzando la distanza Manhattan è possibile definire i seguenti tipi di vicinato che rappresentano quelli più comunemente impiegati nelle operazioni di manipolazione delle immagini tridimensionali:

$$\begin{cases} N_6(X_i) = \{X'(x',y',z') \in Z^3 : D_m(X_i, X') \le 1\} \\ N_{18}(X_i) = \{X'(x',y',z') \in Z^3 : D_m(X_i, X') \le 2\} \cap N_{26}(X_a) \\ N_{26}(X_i) = \{X'(x',y',z') \in Z^3 : \max(D_m(X_i, X')) \le 1\} \end{cases}$$

La stessa definizione può essere data impiegando il concetto di distanza euclidea:

$$\begin{cases} N_6(X_i) = \{X'(x',y',z') \in Z^3 : D_e(X_i, X') \le 1\} \\ N_{18}(X_i) = \{X'(x',y',z') \in Z^3 : D_e(X_i, X') \le \sqrt{2}\} \\ N_{26}(X_i) = \{(x',y',z') \in Z^3 : D_e(X_i, X') \le \sqrt{3}\} \end{cases}$$

Per maggior chiarezza in Figura 3.11 si riporta una descrizione grafica dei vicinati sopra definiti.

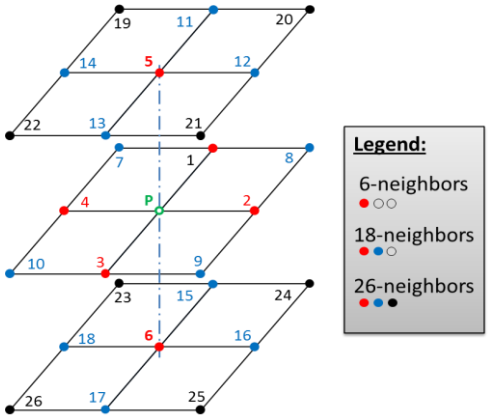
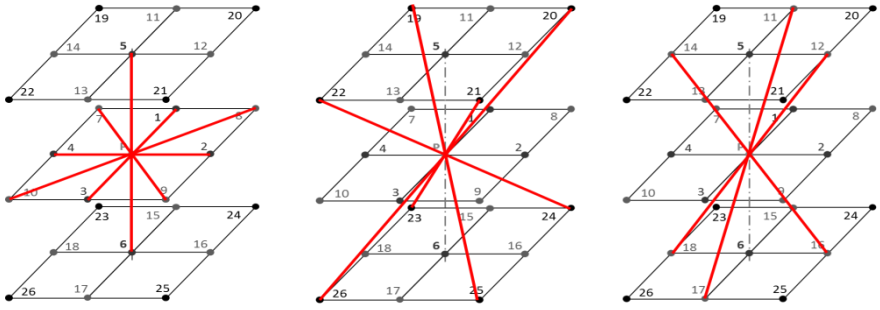


Figura 3.11 – Rappresentazione grafica dei k-vicinati con $k = 6$, $k = 18$ e $k = 26$

Come si può osservare in Figura 3.12, per un dato voxel ω_i è possibile individuare una serie di traiettorie che, attraversando il baricentro P_i del voxel stesso, congiungono i baricentri P_k di ulteriori due voxel confinanti con ω_i.

Risulta così possibile tracciare una serie di traiettorie che contengono il baricentro del voxel ω_i e che si estendono ai baricentri di tutti I voxel facenti parte del vicinato di ω_i.

E' facilmente verificabile che assunto un vicinato di tipo N_k esistono k/2 traiettorie caratterizzate dalle proprietà sopra citate; ad esempio se si esamina il vicinato N_{26} si avranno 13 traiettorie come illustrato in rosso in Figura 3.12.



(a) (b) (c)

Figura 3.12 – Direzioni individuabili nel vicinato N_{26}

Partendo dalle definizioni sopra introdotte è possibile avvalersi del seguente pseudo-codice per definire e archiviare in una matrice S le 13 stime dello spessore (ϑ_i con $i = 1,2,..13$) localmente per ciascun voxel dell'oggetto Ω contenuto nell'immagine 3D. Le direzioni $v_{k/2}$ di esplorazione dell'oggetto rappresentano ciascuna delle traiettorie precedentemente descritte:

Set S as a matrix composed by $\#\Omega$ rows and k/2 columns
For $\omega_i \in \Omega$
  Considering $N_k(P_i)$, find the $v_{k/2}$ vectors
    For each direction *d* defined by $v_{k/2}$
      Set the counters s^+ and s^- equal to 0
      Set the current voxel ω_c equal to ω_i
        For each orientation (+ or -)
          While background is not reached
            Set ω_c as the nearest voxel along the current direction ($v_{k/2}$)
              If ω_c belongs to the foreground and Manhattan distance is used
                Increase by one $s^+(s^-)$
              end
          end

end
If Manhattan distance is used
Set $S(i,d) = s^{+} + s^{-}$
elseif Euclidean distance is used
Set $S(i,d) = D_e(P_i, P_c)$
end
end

Riassumendo, al termine dell'esecuzione dello pseudo-codice si ottiene come risultato una matrice S nella quale risultano memorizzati tutti i valori di spessore stimati ϑ_i per ciascun voxel dell'oggetto e per ciascuna delle direzioni di ricerca (ovvero per ciascuna delle traiettorie).

La metodologia proposta prevede di sintetizzare per ciascun voxel le diverse stime compiute lungo le diverse traiettorie fornendo un solo valore di spessore da associare a ciascun voxel.

A tale scopo per ciascun voxel viene compiuta una operazione di manipolazione dei valori ad esso associati. Supponendo, ad esempio, di aver impiegato un vicinato di tipo N_{26} , per ciascun voxel si avrebbero a disposizione 13 differenti stime dello spessore. La prima operazione che viene condotta riguarda l'ordinamento di tali valori in ordine crescente (vedi Figura 3.13). Le operazioni che seguono sono descritte singolarmente nei paragrafi successivi.

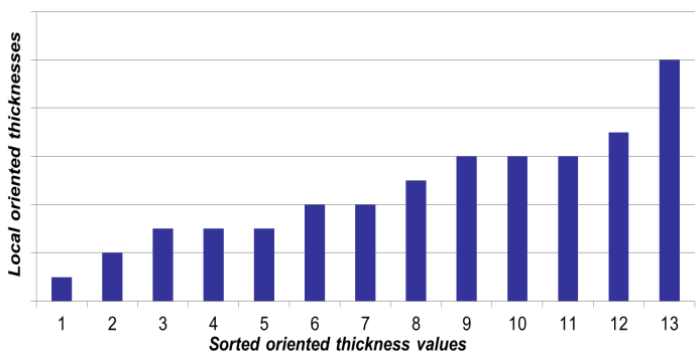


Figura 3.13 – Ordinamento dei valori di stima dello spessore riferiti al singolo voxel

3.2.3.1 Stima dello spessore locale dell'oggetto

Il tredicesimo valore, ovvero il valore più elevato tra quelli associati al voxel, rappresenta la miglior stima possibile della direzione principale, nei dintorni del voxel stesso, dell'oggetto rappresentato nell'immagine. Dato che questa direzione rappresenta localmente la direzione di sviluppo dell'oggetto, essa risulta pressoché ortogonale rispetto alla (minima) sezione dell'oggetto e, pertanto, non deve essere tenuta di conto nella determinazione del valore dello spessore in quel punto. D'altro canto, è possibile affermare che nel caso in cui la direzione principale dell'oggetto coincida esattamente con quella stimata dalla presente metodologia, si avrebbe che 4 traiettorie sulle 13 prese in esame giacerebbero sul piano di sezione. In tale circostanza la loro media darebbe una stima accurata dello spessore dell'oggetto in quel punto. Ad esempio, nel caso mostrato in Figura 3.14 è possibile notare come, essendo la direzione principale stimata con ottima approssimazione, i valori più bassi di spessore possono essere mediati per fornire una stima accurata del valore dello spessore della nuvola in corrispondenza del voxel in esame evidenziato in rosso.

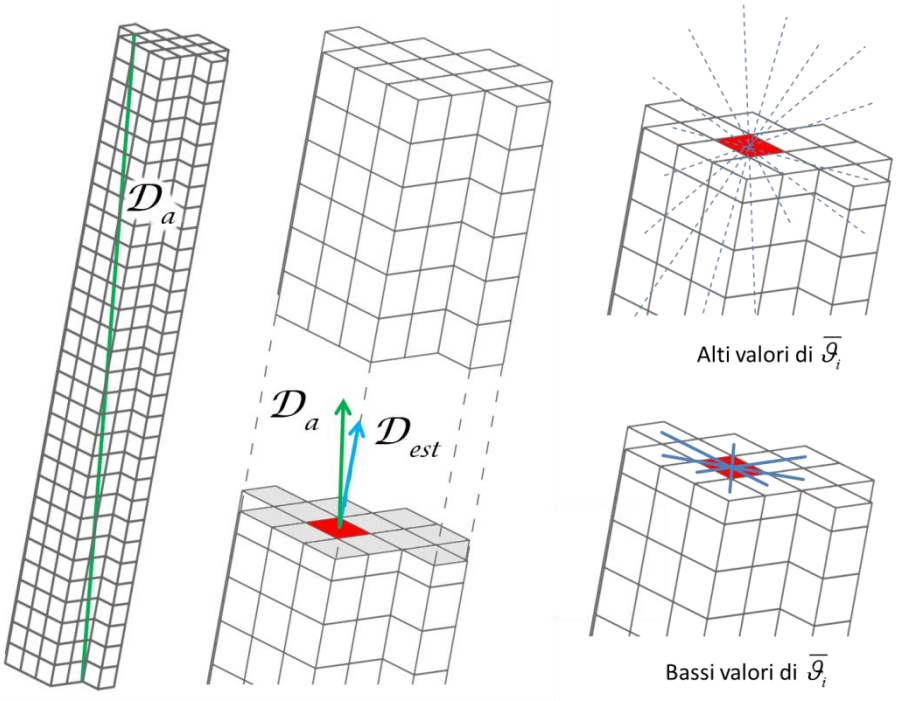


Figura 3.14 – Esempio di individuazione della direzione principale; D_a indica la direzione principale reale, D_{est} quella stimata compiendo valutazioni sulle sole 13 direzioni esaminate.

E' importante notare che più sono le traiettorie esaminate e più alta è la confidenza con cui si stima la direzione principale; di conseguenza più numeroso è il vicinato che si prende in esame, più è accurata la stima dello spessore. Una regola generale per la determinazione dello spessore può quindi essere sintetizzata con la seguente formula:

$$LT_i = \frac{1}{4+\varepsilon} \sum_{j=1}^{4+\varepsilon} \mathcal{Gi}_{sort}(j)$$

dove ε rappresenta un parametro di taratura del modello al variare del numero di traiettorie k/2 che si prendono in esame. Per $k = 26$, ε può essere posto pari a 2. Ovviamente più traiettorie si considerano, ovvero più numeroso è il vicinato preso in considerazione, e più aumenta l'accuratezza della stima dello spessore.

La nota negativa legata all'incremento delle traiettorie esaminate riguarda la complessità computazionale che purtroppo cresce velocemente, basti pensare che passare da un vicinato $3x3x3$ ad un vicinato $5x5x5$ comporta un aumento delle traiettorie da valutare da 26 a 124.

3.2.3.2 Determinazione della probabilità di appartenenza alle zone di intersezione

La procedura di individuazione delle intersezioni prosegue impiegando lo spessore LT_i appena determinato per procedere alla normalizzazione dei restanti valori di spessore non ancora presi in considerazione, ovvero nel caso dell'esempio quelli dal settimo al dodicesimo inclusi (vedi Figura 3.15).

L'elaborazione di questi rimanenti valori permette infatti di valutare se e in quale misura esistono ulteriori direzioni dominanti oltre alla principale; è facile immaginare, infatti, che nel punto di intersezione di due o più entità geometriche non vi sia una unica direzione principale.

L'operazione di normalizzazione serve per omogeneizzare il risultato lungo tutto l'oggetto e renderlo il più possibile indipendente da eventuali possibili variazioni di spessore tra zone diverse dell'immagine.

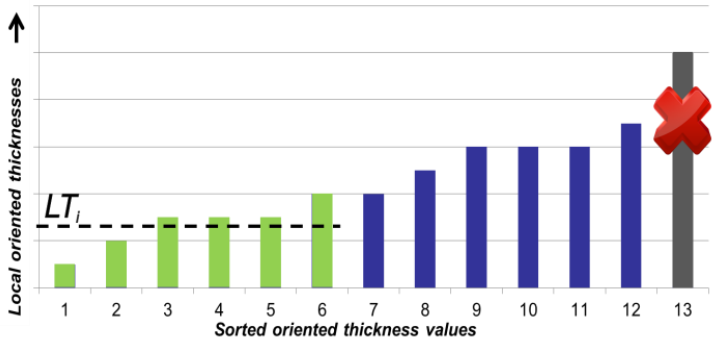


Figura 3.15 – Gestione dei dati relativi agli spessori e determinazione di LT_i

I valori normalizzati vengono, quindi, esaminati in modo da quantificare la probabilità che il voxel in esame si trovi o meno in una zona di intersezione. La quantificazione di tale probabilità si identifica con il risultato dell'integrazione numerica mediante regola del trapezio [38] dei valori stessi.

Graficamente tale risultato può essere illustrato come l'area sottesa all'istogramma dei valori normalizzati; con riferimento alla Figura 3.16 si può vedere come l'area in rosso caratterizza un voxel avente elevata probabilità di appartenenza ad una zona di intersezione mentre l'area verde identifica un voxel avente esattamente le caratteristiche opposte.

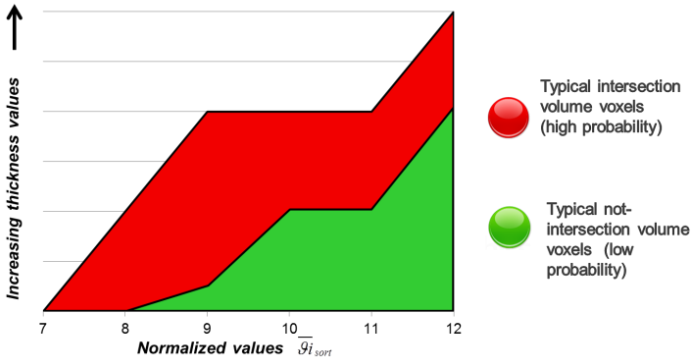


Figura 3.16 – Integrazione numerica e stima della probabilità

3.2.3.3 Individuazione dei voxel appartenenti alle zone di intersezione

La procedura fin qui descritta permette di manipolare i valori determinati per ciascun singolo voxel fornendo come risultato una stima della probabilità per cui il voxel possa o meno appartenere ad una zona di intersezione. Occorre quindi formulare un criterio che consenta di discriminare tra i voxel con elevata probabilità di appartenenza alle zone di intersezione e gli altri.

Per tale scopo la presente metodologia fa uso dello stimatore di Kaplan Meier applicato alla funzione di distribuzione cumulata di tali probabilità; in letteratura tale tecnica è nota come ECDF (*Empirical Cumulative Distribution Function*) [39].

Tale funzione, applicata ai valori dei voxel relativi all'oggetto rappresentato nell'immagine produce una curva sigmoidale, rappresentata a titolo esemplificativo in Figura 3.17 con una curva tratteggiata nera.

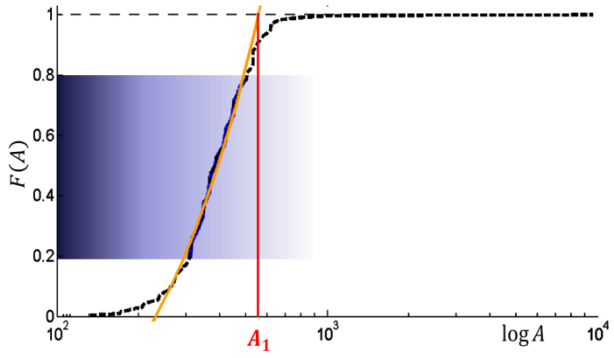


Figura 3.17 – Definizione delle zone di intersezione mediante ECDF

Considerando la sigmoide ricavata, i valori che rappresentano il 60% di probabilità "centrale" della funzione, ovvero tutti quei valori compresi tra il 20% e l'80% di probabilità, vengono sottoposti ad una procedura di regressione lineare che produce la curva rappresentata in arancio in Figura 3.17.

Il valore dell'ascissa corrispondente all'intercetta della linea di regressione e l'ordinata unitaria viene assunto come valore caratteristico della distribuzione ed identificato con A_1.

Tale valore costituisce la base di partenza per la determinazione del valore finale di soglia A_{th} che discrimina tra i voxel appartenenti alle zone di intersezione e tutti gli altri:

$$A_{th} = (1+\alpha)A_1$$

Il parametro α, variabile nell'intervallo compreso tra 0 ed 1, deve essere tarato mediante una serie di test su casi studio diversi.

Utilizzando questo valore di soglia è quindi possibile identificare tutti i voxel appartenenti alle zone di intersezione:

$$F(A_{th}) = \frac{1}{\#\Omega}\sum_{\omega\in\Omega}\{A_\omega \le A_{th}\} \quad (9)$$

E' possibile descrivere la stessa operazione in maniera grafica (vedi Figura 3.18).

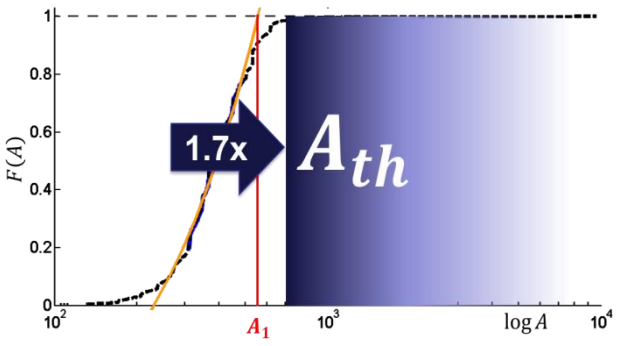


Figura 3.18 – Definizione della soglia ed estrazione dei voxel di intersezione supponendo di porre $\alpha = 0.7$

A titolo di esempio si illustra di seguito un caso studio nel quale il valore di α è stato fatto variare; come si vede in Figura 3.19 nel caso in cui il valore sia troppo basso si ha la presenza di falsi positivi mentre nel caso di un valore troppo elevato (Figura 3.20) si manifesta il problema opposto ovvero la mancata individuazione di alcune zone di intersezione. Solamente un corretto settaggio del parametro α permette di individuare correttamente tutte le zone di intersezione come mostrato in Figura 3.21.

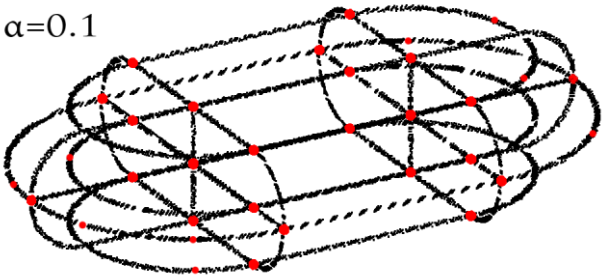


Figura 3.19 – Zone di intersezione individuate per $\alpha = 0.1$

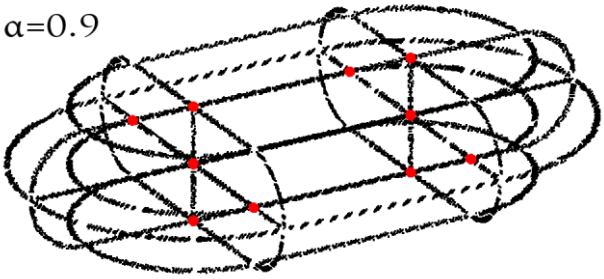


Figura 3.20 – Zone di intersezione individuate per $\alpha = 0.9$

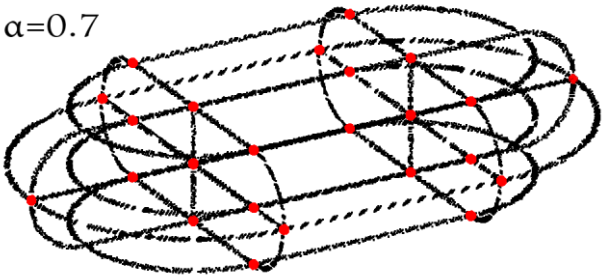


Figura 3.21 – Zone di intersezione individuate per $\alpha = 0.7$

Risulta quindi possibile affermare che il metodo appena esposto, pur presentando alcune lacune nella gestione di geometrie particolarmente complesse, si dimostra efficace nel caso in cui lo "spessore" del modello wireframe raster tridimensionale si mantenga pressoché costante in ogni sua parte. Nel caso in cui ciò non sia verificato è infatti preferibile applicare altre metodologie come sarà in seguito dettagliato nel paragrafo 3.2.5.

3.2.4 Metodo della scheletonizzazione 3D

Un ulteriore metodo sperimentato allo scopo di determinare le zone di intersezione nell'oggetto raffigurato nell'immagine raster 3D è stato quello della scheletonizzazione tridimensionale.

A tale proposito vi sono una quantità di lavori pubblicati in letteratura [37,40–46] anche se dal punto di vista implementativo tali tecniche sembrano ancora ad uno stadio di sviluppo prematuro. Mancano, infatti, quei presupposti di robustezza e generalità che ne consentirebbero un semplice ed immediato impiego come strumento di supporto per metodologie di più ampio respiro quale è la presente.

Tuttavia è necessario evidenziare come le criticità maggiori fossero legate sia a questioni metodologiche che ad aspetti implementativi delle tecniche proposte: la complessità computazionale di tali algoritmi sono ancora troppo elevate per permetterne un impiego privo di limitazioni al caso studiato.

Nel corso del presente lavoro sono stati implementati e testati alcuni tra gli algoritmi sopra citati, il campo di prova di tali implementazioni è stato tuttavia limitato a piccoli esempi composti da immagini aventi bounding box di dimensioni assai limitate e pertanto di scarso interesse per il lavoro qui presentato. Ciò nonostante, dalla sperimentazione effettuata, è stato possibile riscontrare che anche i più evoluti tra gli algoritmi di scheletonizzazione producono risultati piuttosto deludenti, dando origine a geometrie affette da artefatti.

3.2.5 Metodo dello smoothing

L'immagine rappresentante il modello wireframe in formato raster viene, in questo metodo, ricondotta ad un'immagine binaria uguagliando a 1 tutti i voxel con valore diverso da 0. Dal momento che, come sarà chiarito successivamente, i valori precedentemente ottenuti risultano necessari alle successive elaborazioni, questi vengono, comunque, mantenuti in memoria.

L'immagine binaria viene, innanzi tutto, sottoposta ad processo di chiusura morfologica [47] con un kernel unitario di dimensioni $3 \; x \; 3 \; x \; 3$; questa operazione si esegue in modo da correggere eventuali zone non connesse (vuoti) causate dalle operazioni di traduzione nello spazio 3D delle viste bidimensionali di partenza.

La procedura si articola poi in quattro passaggi sfruttando, nella parte finale, una rappresentazione a grafo del modello tridimensionale:

1. estrazione e manipolazione di una mesh di tipo triangolare;
2. individuazione delle zone del modello contenenti possibili nodi;
3. analisi dei possibili vertici e determinazione della lista dei nodi (vertici);

4. individuazione degli archi (curve) componenti il modello.

3.2.5.1 Estrazione e manipolazione di una mesh di tipo triangolare

Il modello binario in ingresso a questa fase di elaborazione necessita di un preprocessamento; la sua rappresentazione iniziale a voxel viene, infatti, tradotta in un modello a mesh triangolare. In particolare si estrae un modello a mesh triangolare che rappresenta il "guscio" dell'oggetto contenuto nell' immagine tridimensionale fornita in ingresso. In altre parole, la struttura a voxel viene elaborata in modo tale da ottenere una mesh chiusa la cui superfice coincide con quella di delimitazione tra i voxel del background e quelli dell'oggetto.

A livello procedurale, l'immagine raster 3D viene esaminata e tutti i voxel facenti parte dell'oggetto raffigurato nell'immagine stessa vengono presi in considerazione; per ciascuno di questi si esegue un controllo avente lo scopo di definire se si sia o meno in presenza di un voxel di superficie (ovvero se il voxel ha nel suo vicinato almeno un altro voxel appartenente al background dell'immagine).

Tale controllo viene effettuato impiegando un algoritmo di tipo "hit and miss" (molto conosciuto e ampiamente impiegato nel processamento delle immagini digitali) utilizzando una serie di maschere tre delle quali (sulle sei totali utilizzate) sono illustrate in Figura 3.22.

A titolo esemplificativo, Il voxel grigio centrale in ognuna delle tre maschere rappresenta il voxel appartenente all'oggetto raster tridimensionale che si intende analizzare; per ciascuno dei suoi sei vicini (ovvero per ciascuno dei voxel trasparenti mostrati in figura) si verifica l'appartenenza o meno al background dell'immagine, nel caso uno o più dei sei vicini appartenessero al background la procedura genera opportunamente la mesh triangolare di delimitazione.

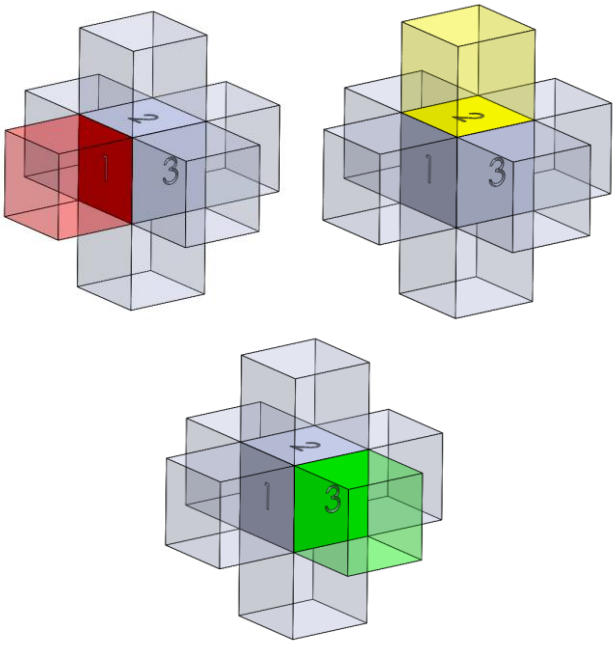


Figura 3.22 – Tre delle sei maschere impiegate nell'algoritmo hit and miss per l'identificazione dei voxel esterni.

Più in dettaglio, sempre riferendosi alla Figura 3.22, nel caso in cui il voxel di colore rosso appartenesse al background dell'immagine, l'algoritmo procederebbe alla generazione di una mesh triangolare sulla faccia "1" (quella evidenziata in rosso) producendo come risultato i triangoli mostrati in Figura 3.23.

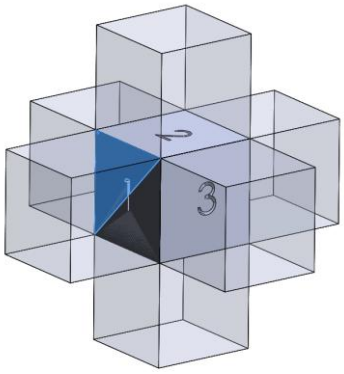


Figura 3.23 – Triangoli di mesh generati sulla faccia "1"

Nel caso in cui più di un vicino appartenesse al background tutte le relative facce del voxel verrebbero trasformate in mesh triangolare; si prendano ancora come riferimento le maschere di Figura 3.22 e sia ponga ad esempio che il voxel rosso e quello verde appartengano entrambi al background, la procedura di generazione della mesh darà come risultato quello mostrato in Figura 3.24.

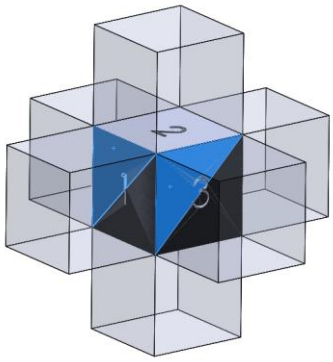


Figura 3.24 – Triangoli di mesh generati sulla faccia "1" e sulla faccia "3"

Ovviamente se nessuno dei sei vicini del voxel in questione appartiene al background dell'immagine l'algoritmo non genera nessuna mesh e prosegue esaminando il voxel successivo. Il risultato finale di questa operazione coinciderà quindi con una mesh di tipo triangolare dove ciascuna faccia di interfaccia tra un voxel dell'oggetto e un voxel del background verrà tradotta in una coppia di triangoli.

Questa fase di pre-processamento in mesh triangolare si è resa necessaria dal momento che, al fine di identificare la topologia dell'oggetto rappresentato dal modello wireframe raster, si è adottata una tecnica di smoothing nota nel mondo del reverse engineering e più in generale della manipolazione di modelli poligonali: lo smoothing Laplaciano [48].

Grazie alla particolare conformazione topologica della mesh sottoposta ad operazione di smoothing non occorre controllare con troppa attenzione il numero di iterazioni da eseguire sull'oggetto. Come noto, infatti, l'operazione di smooothing su mesh di tipo generico può portare alla degenerazione della mesh stessa e alla conseguente impossibilità di trarre informazioni ad operazione completata.

Con riferimento alla Figura 3.25 è infatti possibile osservare come una mesh (la più grande in figura) sottoposta iterativamente ad operazioni di smoothing successive (in figura si evidenziano due stadi di smoothing) tenda a degenerare e, al limite, a collassare in un punto.

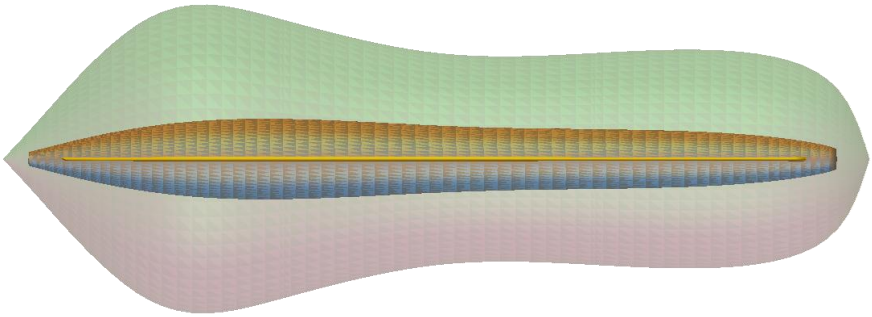


Figura 3.25 – Applicazione dell'operazione di smoothing ad una mesh generica

Nel caso in questione, tuttavia, le mesh sottoposte al processo di smoothing sono caratterizzate da forme chiuse il che impedisce la degenerazione e la perdita di dati di tipo topologico anche dopo l'applicazione di numerose iterazioni di smoothing. In Figura 3.26 si mostra un dettaglio di una mesh prima dell'operazione di smoothing, ad uno stato intermedio ed infine dopo il completamento di tutte le iterazioni previste. Come si vede chiaramente dalla figura, l'immagine di sinistra rappresenta la mesh ottenuta dalla conversione di un'immagine tridimensionale; da notare che i nodi della mesh ottenuta risultano avere una configurazione particolare per cui le loro coordinate hanno tutte valori ad una cifra decimale; una traslazione del sistema di riferimento pari a mezza unità nelle tre direzioni principali porta la mesh ad avere nodi che in partenza possiedono tutti coordinate intere.

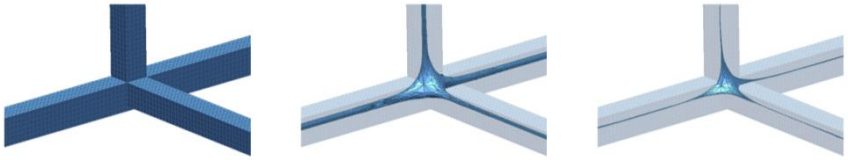


Figura 3.26 – Applicazione dell'operazione di smoothing

3.2.5.2 Individuazione delle zone del modello contenenti possibili nodi

La mesh elaborata con il procedimento di smoothing iterativo introdotto nel paragrafo precedente presenta alcune caratteristiche molto evidenti e, allo stesso tempo, molto utili alla localizzazione delle zone della mesh nelle quali, con elevata probabilità, si andranno a collocare i vertici del modello che si intende ricostruire. Come risulta evidente anche nell'immagine di destra della Figura 3.26, la mesh ottenuta è caratterizzata da un elevato numero di elementi di tipo "spike", ovvero da elementi con un fattore di forma fortemente sbilanciato verso zero.

Definendo infatti tale valore come l'inverso dell'aspect ratio, ovvero, con riferimento alla Figura 3.27, il rapporto tra l'ipotenusa c e l'altezza h relativa all'ipotenusa stessa, è possibile definire un valore variabile nell'intervallo $(0,1]$ per ciascuno dei triangoli costituenti la mesh.

Allo scopo di isolare i triangoli con un fattore di forma più prossimo all'unità (ovvero i triangoli caratterizzati da angoli interni sostanzialmente omogenei tra loro) si è stabilito un valore di soglia, pari a $0{,}5$. I triangoli contraddistinti da un valore di fattore di forma compreso nell'intervallo $[0{,}5;\ 1]$ identificano le zone della mesh nelle quali dovranno essere posizionati i vertici del modello wireframe definitivo.

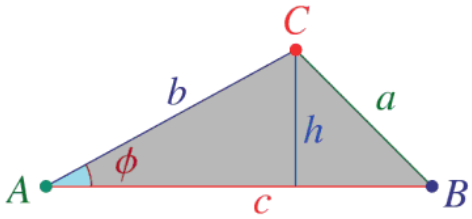


Figura 3.27 – Dimensioni caratteristiche del triangolo.

Così facendo si genera, infatti, una serie di cluster dislocati lungo la mesh che identificano possibili collocazioni dei vertici finali.

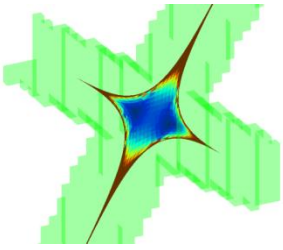


Figura 3.28 – Fattore di forma dei triangoli della mesh elaborata; il colore blu identifica triangoli con fattore di forma superiore a 0,5.

L'analisi di tali gruppi di triangoli procede con la determinazione di tutti i baricentri dei singoli cluster, così facendo si crea una "lista" di cluster che rappresentano una prima versione dei vertici dell'oggetto.

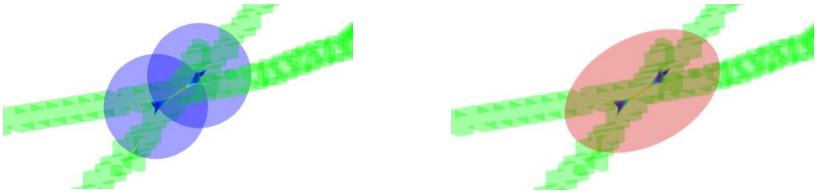
Si passa, quindi, all'analisi dei vari cluster definendo una nuova entità: la sfera di taglio. In altre parole si stabilisce un ulteriore parametro, il raggio di una sfera, che consente di stabilire se due o più cluster sono tra loro indipendenti oppure condividono una parte della loro sfera di taglio.

Il parametro del raggio viene, nel presente lavoro, impostato empiricamente sulla base dello "spessore medio" delle entità raster caratteristiche dell'oggetto rappresentato nell'immagine 3D originale. In questa fase un'alternativa di stima del parametro raggio può consistere nello stabilire un valore che sia funzione dello spessore delle linee nell'immagine 2D acquisita inizialmente e del numero di iterazioni di dilatazione che si sono eseguite per poter produrre un'immagine 3D topologicamente corretta.

Il calcolo dello spessore medio viene effettuato mediante considerazioni di tipo empirico sullo spessore delle entità appartenenti alle immagini bidimensionali di partenza e sulla quantità di iterazioni di dilatazione occorse per determinare una struttura raster 3D valida.

Mediante l'analisi delle sfere di taglio è quindi possibile determinare eventuali intersezioni tra cluster diversi; l'individuazione di tali circostanze porta all'unione dei cluster coinvolti. In tal caso si procede, inoltre, alla ridefinizione di un volume di taglio del nuovo cluster questa volta di tipo ellissoidale; gli assi dell'ellissoide vengono orientati grazie alla PCA (Principal Component Analysis) [49] effettuata sui nodi dei triangoli dei cluster che si debbono raggruppare. Per quanto riguarda le dimensioni dell'ellissoide, la lunghezza dell'asse minore viene imposta pari al doppio del raggio della sfera di taglio di partenza, mentre per la lunghezza dell'asse maggiore si ricorre al rapporto tra gli autovalori forniti dalla PCA.

In Figura 3.29 si mostra il risultato dell'applicazione della tecnica sopra descritta nel caso di intersezione tra sfere di taglio. Al termine della procedura si ottiene l'ellissoide di Figura 3.29b i due cluster originali di Figura 3.29a vengono cancellati dalla lista.



(a) (b)

Figura 3.29 – Esempio di due cluster che condividono parte della loro sfera di taglio

L'individuazione dell'intero insieme di sfere ed ellissoidi permette di separare l'immagine raster iniziale in un numero di oggetti semplici (Figura 3.30) pari al numero di entità geometriche che comporranno il modello wireframe finale.

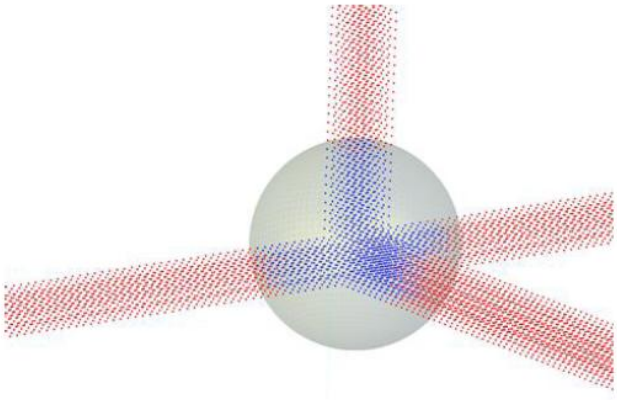


Figura 3.30 – Esempio di cluster (evidenziato in blu) la cui rimozione da luogo a tre sotto-immagini relative ad altrettanti entità geometriche 3D (evidenziate in rosso)

A tale scopo si creano delle immagini ausiliare composte dalle singole sfere o ellissoidi rasterizzati. In particolare ogni cluster viene rappresentato per mezzo di un'immagine raster 3D binaria avente lo stesso bounding box dell'immagine 3D originale ed avente valore unitario soltanto in corrispondenza dei voxel rappresentanti la sfera o l'ellissoide.

Successivamente si assegna, tramite una semplice operazione di moltiplicazione scalare, un valore diverso a ciascun cluster in modo tale da "etichettare" ogni immagine creata. In altre parole le immagini binarie dei cluster vengono differenziate tra di loro in base ad un numero progressivo intero che si sostituisce al valore unitario tipico delle immagini binarie. L'elemento che contraddistingue il volume occupato dalla sfera o dall'ellissoide assume pertanto valore pari a uno nel primo cluster, due nel secondo, tre nel terzo e così via fino ad assumere valore n per l'n-esimo cluster.

A questo punto della procedura si ha quindi una struttura dati composta dai seguenti oggetti:

1- un'immagine tridimensionale W rappresentante il modello wireframe ottenuto grazie all'estrusione delle viste originali bidimensionali;
2- tante immagini $\mathrm{C_i}$ quanti sono i cluster (della stessa dimensione dell'immagine di cui al punto primo) contenenti valori non nulli solo

e soltanto in corrispondenza del volume del cluster che rappresentano.

Sulla base di questi dati si procede come segue per:

1- separare le varie entità raster che daranno origine alle curve del modello finale;
2- mantenere l'informazione topologica che andrebbe persa a causa della separazione di tali entità.

Si genera un'immagine $C = \sum C_i$, si rielaborano singolarmente le varie immagini C_i operando su di esse una dilatazione unitaria che le trasformerà in C_i^* (in questa operazione l'immagine binaria deve mantenere l'elemento non nullo al valore iniziale) e si crea una seconda immagine $C^* = \sum {C_i}^*$.

Grazie alle immagini sopra prodotte è quindi poi possibile definire un'ulteriore operazione andando a creare l'immagine I^*:

$$I^* = C^*(I - C) \qquad (3.3)$$

Con riferimento alla Figura 3.31 (rappresentata per semplicità in due sole dimensioni), siano A, B e C tre diversi tratti dell'oggetto raffigurato nell'immagine raster e convergenti in una zona X (cluster); quanto descritto nei precedenti capoversi ha come scopo la separazione reciproca di A, B e C mantenendo tuttavia l'informazione riguardo al fatto che tutti e tre condividono la stessa parte terminale (ovvero il cluster X).

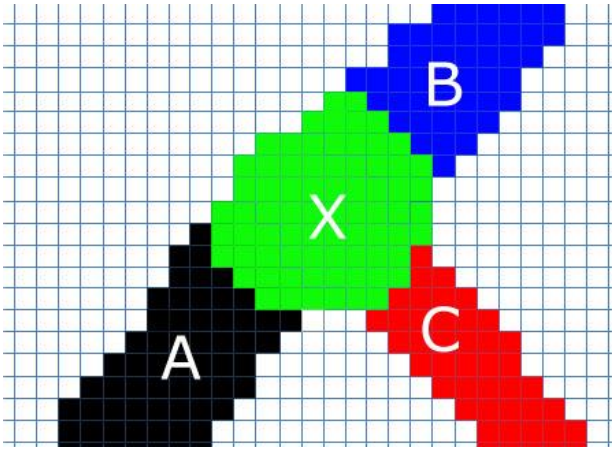


Figura 3.31 – Modello bidimensionale della struttura 3D cluster/entità raster

Grazie all'operazione di dilatazione unitaria del cluster X si genera X* e la differenza tra X ed X* consiste nei voxel evidenziati in arancio in Figura 3.32.

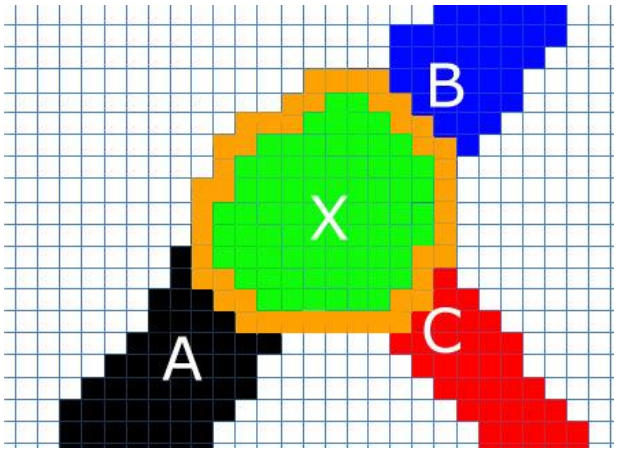


Figura 3.32 – Risultato del processo di dilatazione unitaria del cluster "X"

A questo punto tramite l'applicazione della formula 3.3 si ottiene il risultato finale in due semplici passaggi:

1. sottraendo all'immagine iniziale l'immagine dei cluster si ottiene il risultato illustrato in Figura 3.33;
2. moltiplicando tale risultato per il cluster X* ovvero per il cluster dilatato si produce l'immagine di Figura 3.34.

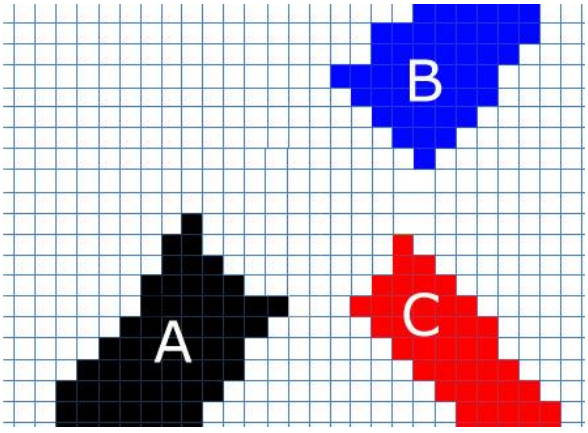


Figura 3.33 – Risultato dell'operazione (I-C) dell'eq. 3.3

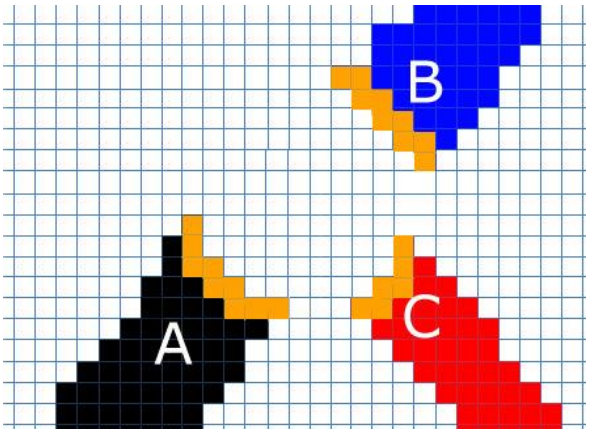


Figura 3.34 – Immagine finale con entità separate corredate dal dato sul cluster di appartenenza

L'immagine di Figura 3.34 (parte di I* dell'equazione 3.3) conterrà quindi le varie entità da ricostruire tra loro separate e corredate dal dato riguardante il cluster su cui confluivano prima di essere separate (tale informazione si rivelerà quindi fondamentale per la ricostruzione della corretta struttura topologica del modello vettoriale finale).

Il metodo dello smoothing qui descritto è stato quello che si è rivelato più efficace sia in termini di robustezza (falsi negativi e falsi positivi teoricamente azzerati), che in termini di generalità dei casi su cui può essere applicato. Un vantaggio senza dubbio importante rispetto alle altre tecniche è quello di lavorare con una mesh che rappresenta il "guscio" dell'oggetto e non con ciascun singolo voxel dell'oggetto; questa caratteristica permette l'esecuzione della procedura con una quantità di dati assai minore rispetto alle altre tecniche descritte.

E' quindi possibile, a questo punto della procedura, estrarre i singoli oggetti dall'immagine ed analizzarli separatamente e localmente mantenendo, inoltre, l'informazione necessaria e sufficiente ad un reinserimento degli stessi, una volta vettorializzati, nell'immagine globale.

3.3 Generazione dei vertici finali

Una volta determinata la topologia del modello contenuto nell'immagine raster tridimensionale è stato quindi possibile separare, come descritto nel paragrafo precedente, le varie entità geometriche pur ancora sotto forma di "oggetti" raster tridimensionali.

Il primo passaggio che si affronta avendo a disposizione lo schema topologico del modello è quello di collocare i vertici definitivi in posizione ottimale, compatibilmente con le informazioni a disposizione. A tale scopo, in questa fase preliminare della metodologia, si procede ad una prima elaborazione delle singole entità raster separate. In particolare, si procede ad un'operazione di valutazione della posizione dei vertici per mezzo dell'inserimento di una prima versione di curve 3D che descrivano, con "sufficiente approssimazione", la geometria finale.

Va infatti notato come la ricerca della posizione dei vertici all'interno delle zone di taglio possa essere determinata, con ottima approssimazione, tenendo conto delle direzioni delle curve incidenti sul vertice stesso.

La prima operazione da eseguire al fine di estrarre i vertici di primo tentativo è la "pulitura" dell'immagine tridimensionale. Questa pulitura consiste nell'eliminazione di tutte le entità per le quali la definizione, in termini di voxel contenuti, non è sufficiente, o quanto meno non affidabile, per procedere alla ricostruzione dell'entità vettoriale relativa.

In altre parole ciascuna entità deve mostrare una direzione prevalente di propagazione per essere ritenuta valida; una entità nella quale non vi siano direzioni prevalenti di sviluppo risulta di difficile gestione, infatti su di essa sarebbe problematica qualsiasi operazione di fitting (risulta infatti difficile pensare ad una curva in grado di "fittare" con buona qualità una serie di punti disposti in un volume sferoidale, cioè non disposti lungo una direzione principale).

L'entità della "pulizia" dipende ovviamente dalle caratteristiche dell'immagine in questione; una buona linea guida da seguire è quella, suggerita dall'esperienza maturata, di impostare una soglia di voxel minima (perché un oggetto sia considerato) proporzionale allo spessore locale dell'immagine raster 3D del modello wireframe (in pratica lo stesso concetto applicato nella determinazione del raggio delle sfere di taglio).

3.3.1 Estrazione delle traiettorie

La posizione dei vertici all'interno delle zone di taglio viene quindi determinata previa identificazione delle traiettorie delle curve che vi incidono; a tal proposito ciascun oggetto necessita di essere esaminato e approssimato con una curva.

L'idea che sta alla base del metodo ideato è la seguente:

1- per ciascun oggetto raster si definisce una polilinea che approssimativamente percorre l'oggetto lungo la sua direzione principale di sviluppo (linea verde in Figura 3.35);
2- si interpola ciascuna polilinea con una curva, nel caso specifico di tipo spline cubica (linea rossa in Figura 3.36);

3- si ordinano i voxel dell'oggetto in funzione della proiezione del loro baricentro sulla curva spline;
4- si procede al "fitting" dei baricentri dei voxel ordinati tramite una qualunque tecnica di fitting pesato tridimensionale (curva blu in Figura 3.37); i pesi si ricavano dal valore dei voxel stessi nell'immagine iniziale che si ricorda essere di tipo gray-scale.

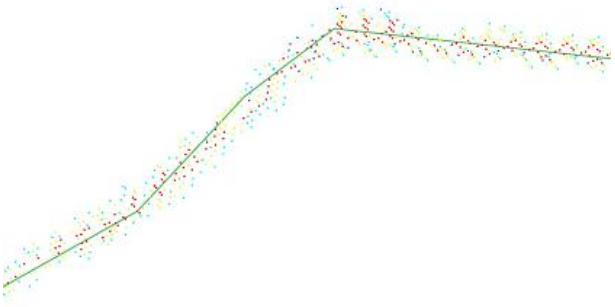


Figura 3.35 – Polilinea iniziale per l'ordinamento dei voxel

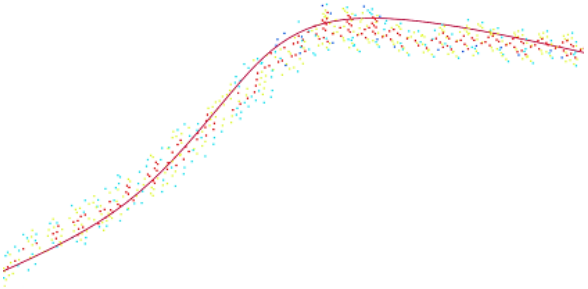


Figura 3.36 – Curva interpolante sui nodi della polilinea iniziale

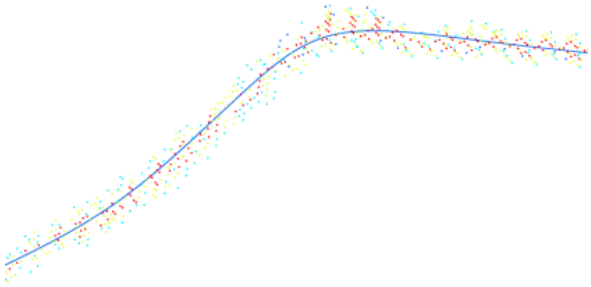


Figura 3.37 – Curva finale approssimante

Il risultato di questa operazione di approssimazione degli oggetti con curve di tipo spline fornisce esattamente il risultato desiderato ovvero permette di definire con una buona accuratezza le traiettorie delle curve che compongono il modello wireframe fino ad ora caratterizzato da informazioni di tipo raster.

Di seguito vengono illustrati i principali passaggi che portano alla determinazione delle curve approssimanti.

3.3.1.1 Costruzione della polilinea iniziale

La prima parte della procedura consiste nel costruire una polilinea che approssimativamente attraversi le singole entità raster isolate. A tale scopo viene impiegata una tecnica basata sull'utilizzo congiunto di due strumenti, il primo è l'Euclidean Minimum Spanning Tree (EMST) [50] ed il secondo è nuovamente la PCA.

E' da notare che il metodo descritto risulta di amplia applicabilità. In particolare, il metodo qui proposto è applicabile a nuvole di punti tridimensionali; la sua applicazione a immagini, pur tridimensionali, ne costituisce un caso particolare in cui la "nuvola" ha la particolarità di avere punti a coordinate intere.

Facendo riferimento alla Figura 3.38 è possibile definire in modo casuale un punto dell'immagine di partenza dal quale propagare la polilinea; questo punto è indicato in figura con il simbolo $b_{h,k}$.

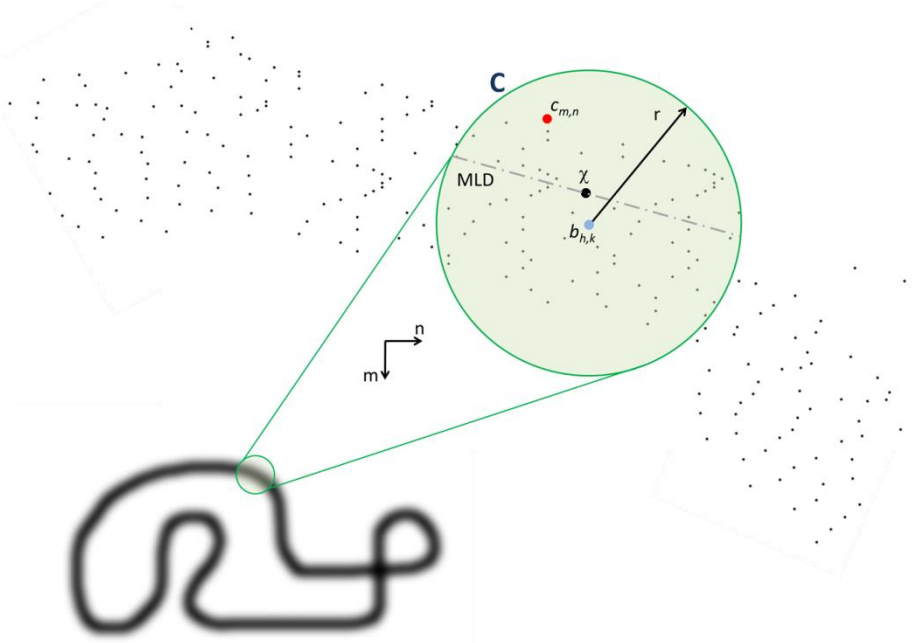


Figura 3.38 – Origine della polilinea iniziale

Il punto individuato permette di definire un dominio di ricerca, nel caso bidimensionale in figura rappresentato da un cerchio, di dimensioni note (imposte); nel caso di una immagine tridimensionale possiamo ricorrere all'uso di un dominio di tipo sferico identificato dal solo valore del raggio della sfera.

Il passaggio immediatamente successivo prevede il calcolo del baricentro di tutti i voxel compresi all'interno di tale dominio, questa operazione dà origine ad un nuovo punto identificato con la lettera χ in Figura 3.38. A questo punto si conduce una PCA sui voxel compresi nel dominio determinando gli assi principali di inerzia del set di voxel in questione. L'asse dominante viene quindi denominato Mean Local Direction (MLD). Sfruttando la direzione della MLD ed il passaggio per il punto χ si costruisce un segmento di retta delimitato dal confine del dominio di ricerca circolare (sferico nello spazio 3D).

Partendo dai punti estremi di questo segmento di definiscono due nuovi domini di ricerca (vedi Figura 3.39), C_{p1} e C_{f1}, aventi i centri in corrispondenza dei due estremi del segmento; a loro volta queste due nuove regioni porteranno alla determinazione di due nuovi baricentri, χ_{p1} e χ_{f1}, e di due nuove MLDs.

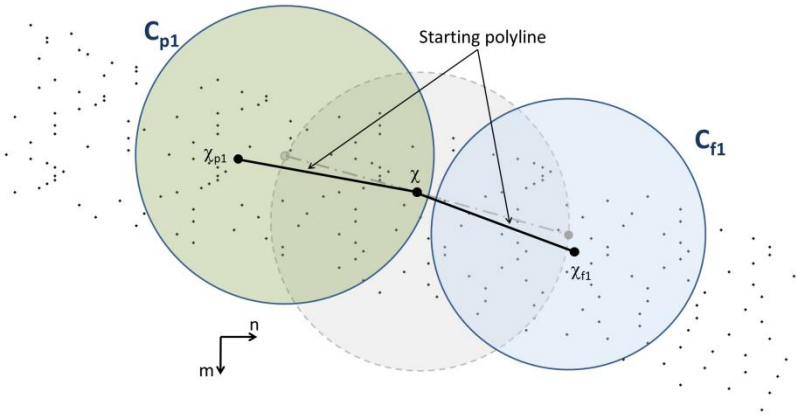


Figura 3.39 – Origine dei primi due tratti della polilinea iniziale

Una volta determinati i due nuovi baricentri si procede alla costruzione dei primi due tratti di polilinea come indicato in Figura 3.39. La procedura viene quindi iterata fintanto che non si giunge alla determinazione di domini di ricerca costituiti da insiemi vuoti.

La polilinea così determinata potrebbe essere utilizzata per il proseguimento della procedura, in realtà vi è la necessità di scongiurare la non corretta identificazione della topologia del tratto di voxel da ricostruire; in altre parole vi è la necessità di verificare che i voxel contenuti all'interno dei domini di ricerca abbiano tutte le carte in regola per appartenervi.

Con riferimento alla Figura 3.40 si può osservare come il dominio di ricerca può, in ragione della sua conformazione geometrica, includere voxel che non appartengono alla stessa zona di nuvola in esame. Al fine di non considerare voxel eccessivamente lontani fra loro, è stato impiegato il EMST.

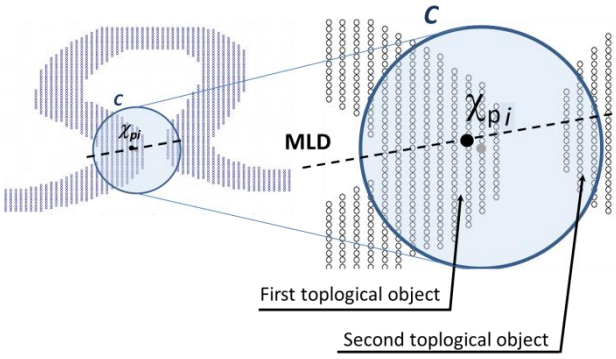


Figura 3.40 – Importanza del EMST

In pratica il EMST fornisce un grafo in cui i nodi sono costituiti dai punti (voxel) mentre gli archi sono generati in modo tale da definire una struttura singola ovvero monoconnessa (grafo) caratterizzata dal fatto che la somma delle lunghezze degli archi stessi sia la minima possibile. In atri termini il EMST consiste in una "rete" di segmenti che:

1. permettono di legare assieme tutti i punti in esame facendo si che da ciascun punto sia possibile raggiungerne un qualsiasi altro;
2. è caratterizzata dal fatto di essere la più "corta" tra tutte quelle realizzabili.

La mancanza di questo controllo potrebbe portare a situazioni come quella rappresentata in Figura 3.41 dove si vede chiaramente che la polilinea non segue la logica con cui sono disposti i voxel ma "salta" da una zona all'altra della nuvola.

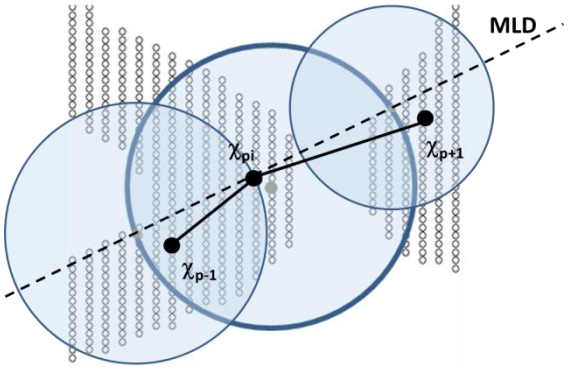


Figura 3.41 – Errore dovuto all'assenza di controllo tramite EMST

Da un punto di vista globale questo tipo di errore produce dei risultati del tipo mostrato in Figura 3.42.

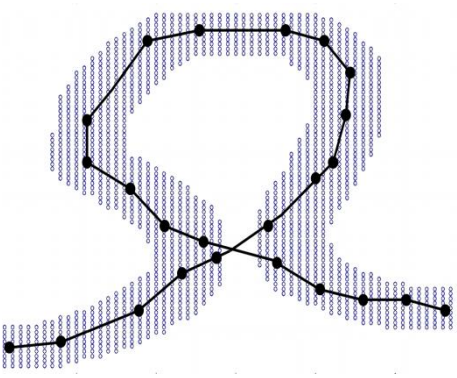


Figura 3.42 – Errata costruzione di una polilinea

L'impiego dello strumento EMST permette di individuare questo tipo di errore e di correggerlo alla radice evitando di inserire nel dominio di ricerca voxel non congrui con la zona di immagine che si sta analizzando; in altre parole l'utilizzo dell'EMST permette di passare dalla situazione illustrata in Figura 3.41 a quella mostrata in Figura 3.43.

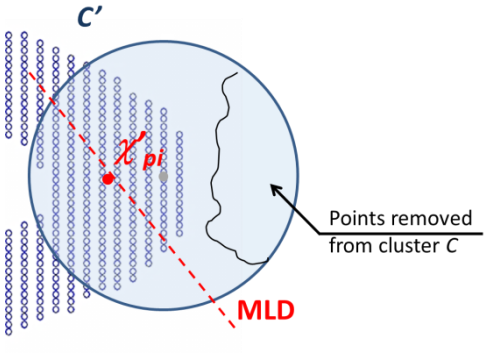


Figura 3.43 – Corretta costruzione della polilinea

Con l'aiuto della Figura 3.44 è possibile dare una ulteriore versione circa l'utilità dell'impiego dell'EMST, in tale figura l'EMST è il percorso rappresentato dai segmenti che collegano i vari punti, grazie alla loro definizione è possibile affermare che il punto evidenziato in verde e quello evidenziato in blu distano ben 12 "passi" lungo tale percorso; in figura si evidenziano per entrambi i punti i loro vicinati di ordine 3, come si evince dall'illustrazione, benché la distanza euclidea che separa il punto verde ed il punto blu è molto limitata e confondibile con le dimensioni dei

due vicinati, questi ultimi possono essere facilmente definiti grazie ad un semplice ragionamento basato sull'EMST. Una volta in prossimità del punto verde sarà semplice escludere dal dominio del calcolo della MLD i punti non "congruenti", basterà infatti una valutazione aggiuntiva a quella iniziale di tipo geometrico (che ad esempio comporterebbe anche l'inclusione di alcuni punti blu); valutando anche la distanza tra i punti del dominio lungo l'EMST è possibile escludere i punti che non possono ragionevolmente far parte del dominio di calcolo della MLD.

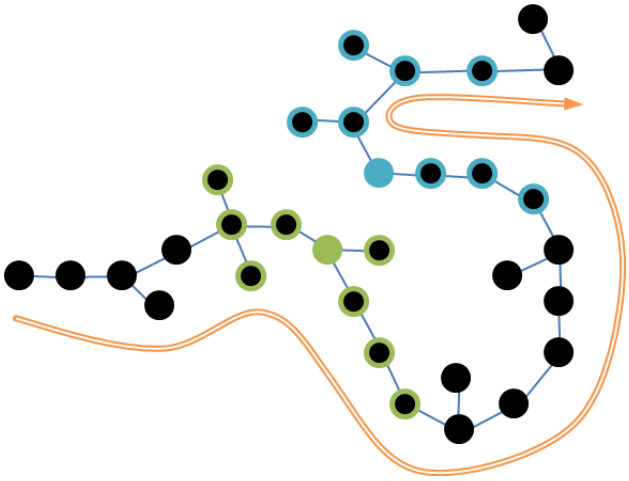


Figura 3.44 – Soluzione all'errore dovuto all'assenza di controllo tramite EMST

Questo strumento permette quindi di ottenere come risultato finale la polilinea corretta (vedi Figura 3.45).

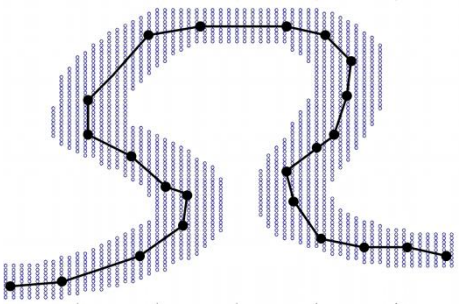


Figura 3.45 – Corretta costruzione della polilinea

3.3.1.1.1 Ordinamento dei voxel e approssimazione con curve di tipo spline

Una volta determinata la polilinea, la procedura prevede la costruzione di una curva di ordinamento di tipo spline; tale curva interpola tutti i nodi della polilinea e rappresenta uno strumento ottimale per l'ordinamento dei voxel su cui si intende costruire la geometria finale. Le sue proprietà di continuità di derivata rispetto alla polilinea iniziale le permettono di identificare in modo univoco l'ordine con il quale i voxel debbono essere considerati nella fase di fitting finale.

Come già anticipato, tale ordine viene determinato seguendo quello con cui i voxel stessi si proiettano sulla curva di ordinamento.

Con i voxel ordinati è quindi possibile procedere all'approssimazione del tratto di immagine 3D con una seconda curva di tipo spline; i metodi impiegati per l'approssimazione per effettuare questo passo possono essere vari; in questa sede si è deciso di ricorrere a spline cubiche approssimanti nel senso dei minimi quadrati.

3.3.2 Posizionamento dei vertici

Una volta ultimato il processo di approssimazione degli oggetti raster con curve spline è necessario studiare le zone in cui queste entità vettoriali convergono per dare un collocamento definitivo ai vertici del modello finale; a tale scopo sono state utilizzate congiuntamente sia le curve sopra determinate sia le zone di taglio individuate grazie all'operazione di smoothing della mesh del modello wireframe raster. In particolare, per ciascuna delle zone di taglio si raccolgono le curve aventi uno dei due endpoint nella prossimità di tale zona e si prolungano andando ad inserire, a partire dall'endpoint stesso, un segmento tangente alla curva e di lunghezza sufficiente ad attraversare la zona di taglio (questo valore di lunghezza viene calcolato per le zone sferiche pari a circa il diametro della sfera e per le zone ellissoidali pari a circa la lunghezza dell'asse principale dell'ellissoide).

Ovviamente, una volta generati tutti i segmenti, non sarà possibile identificare un punto interno alla zona di taglio in cui si abbia il passaggio contemporaneo di tutte le curve (vedere Figura 3.46); per sopperire a questa problematica, la soluzione qui proposta comporta la ricerca del baricentro del voxel appartenente alla zona di taglio la cui posizione sia tale per cui la somma dei quadrati delle distanze dai vari segmenti sia la minima. Si capisce che ciò limita il dominio di ricerca ai soli baricentri dei voxel della zona di taglio, questa soluzione permette di ottenere tuttavia un rapporto tra costo computazionale ed accuratezza piuttosto vantaggioso, decisamente soddisfacente per gli obiettivi che si propone di affrontare la presente metodologia.

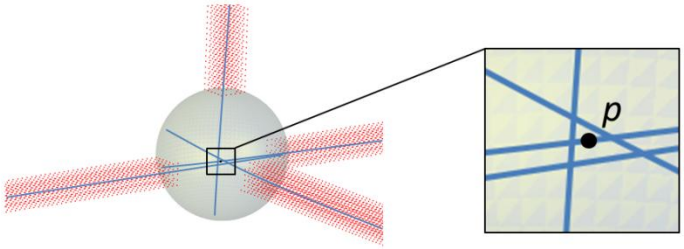


Figura 3.46 – Posizionamento dei vertici finali

A supporto di quanto detto sopra si pensi quale sia l'influenza di un errore pari ad un pixel rapportata all'intero processo di ricostruzione; nella maggior parte degli esempi che verranno proposti nel capitolo dei casi studio le viste in ingresso sono state acquisite con risoluzioni tali da descrivere lo spessore del tratto di disegno tramite almeno 3 o 4 pixel. Si capisce, quindi, come l'errore che si commette sia totalmente trascurabile. Un'ultima commento riguarda la descrizione dei limiti del dominio di ricerca per i vertici finali: come detto, il dominio è fatto coincidere con l'insieme dei voxel contenuti nella zona di taglio e sarebbe un errore estendere ulteriormente tali limiti che di fatto assumono un valore concettuale molto forte. Infatti, solamente tali punti sono stati originati dall'intersezione delle viste ortogonali di partenza; uscire da tale dominio significherebbe permettere di individuare come vertici punti la cui proiezione non ricade sui tratti di disegno nelle tre viste originali.

3.4 Generazione delle curve finali

Grazie all'individuazione della corretta posizione dei vertici risulta finalmente possibile determinare le curve che meglio approssimano il modello wireframe di partenza. Il precedente "fitting" di curve spline non era in grado di fornire una rappresentazione corretta proprio in ragione della mancanza dei dati sui vertici finali. La differenza principale, quindi, risiede proprio nel fatto che in questo caso l'approssimazione avviene risolvendo un problema "vincolato" ovvero imponendo a ciascuna curva il passaggio dai vertici che la delimitano.

Dal punto di vista procedurale il tutto si riduce alla soluzione di un problema di ottimizzazione vincolata.

Nello specifico è stato utilizzato un algoritmo a punto interno [51–53] dal momento che si è dimostrato, nella totalità dei casi studio analizzati, capace di garantire risultati più che soddisfacenti. Per quanto riguarda la funzione obbiettivo da minimizzare si è scelto un approccio di tipo energetico, valutando sia l'errore potenziale sia l'errore di tipo elastico:

$$E = E_p + E_e \tag{3.4}$$

Circa l'errore potenziale, è stato adottato un metodo di calcolo che consentisse di giungere rapidamente a convergenza e di affinare, successivamente, la soluzione nelle ultime iterazioni. In particolare il calcolo dell'errore potenziale è stato implementato con una funzione che assume un comportamento condizionale:

1. se il valore dell'errore è sopra una certa soglia, l'errore viene calcolato con il metodo di seguito descritto come metodo delle "funi",
2. se l'errore scende sotto la soglia imposta si passa ad una valutazione di tipo "classico" misurando l'errore in termine di somma dei quadrati delle distanze dei punti da approssimare dalla curva approssimante.

Nelle equazioni 3.5 e 3.6 vengono presentati rispettivamente il metodo di valutazione dell'errore potenziale per mezzo del metodo delle funi e per mezzo del metodo "classico"; in entrambi i casi il parametro n è pari al numero dei punti su cui si valuta l'errore di fitting mentre f_i rappresenta l'errore con il metodo delle funi per il punto i-esimo e d_i rappresenta l'errore con il metodo classico per il punto i-esimo.

Il valore di soglia E_p^{th} nell'equazione 3.7 deve essere individuato mediante campagna di prove sperimentale.

$$E_{pf} = \sum_{n}^{i=1} f_i \tag{3.5}$$

$$E_{pd} = \sum_{n}^{i=1} d_i \tag{3.6}$$

$$E_p = \begin{cases} E_{pf} \; con \; E_p \geq E_p^{th} \\ E_{pd} \; con \; E_p < E_p^{th} \end{cases} \tag{3.7}$$

Il metodo delle "funi" è stato pensato ed implementato per permettere al solutore di muoversi verso un punto di convergenza della soluzione e, possibilmente, di farlo anche in modo rapido.

In Figura 3.47 viene illustrato l'utilizzo combinato delle due tipologie di valutazione dell'errore sopra citate. Inizialmente, come si vede dalla figura, l'entità approssimante consiste in un segmento di spline rettilineo e lo scopo è quello di approssimare dieci diversi punti (punti blu in figura); dalla spline si estraggono quindi tanti valori equispaziati del parametro quanti sono i punti da approssimare (in questo caso dieci). Sfruttando l'ordinamento dei punti (effettuato al momento del primo fitting) si creano una serie di distanze che legano il primo punto alla coordinata della spline relativa al primo valore del parametro e così via (nel caso illustrato si creeranno dieci valori di distanza, rappresentati in figura dai segmenti arancioni ovvero dal parametro f_i nell'equazione 3.5 con $i = 1, \dots, 10$).

Nei vari step di approssimazione si cerca di modellare la curva minimizzando la somma dei quadrati di tali valori delle distanze; esisterà una iterazione in cui il valore dell'errore scende sotto una soglia prestabilita, a quel punto la curva comincerà ad avere una forma geometrica che seppure grossolanamente seguirà l'andamento dei punti da approssimare. In questo istante la valutazione dell'errore mediante il metodo delle funi viene abbandonata e si passa ad una vera e propria approssimazione ai minimi quadrati dove l'errore viene valutato per mezzo del calcolo della distanza euclidea tra ciascun punto e la curva (segmenti in verde in Figura 3.47 ovvero parametro d_i nell'equazione 3.6 con $i = 1, \dots, 10$)).

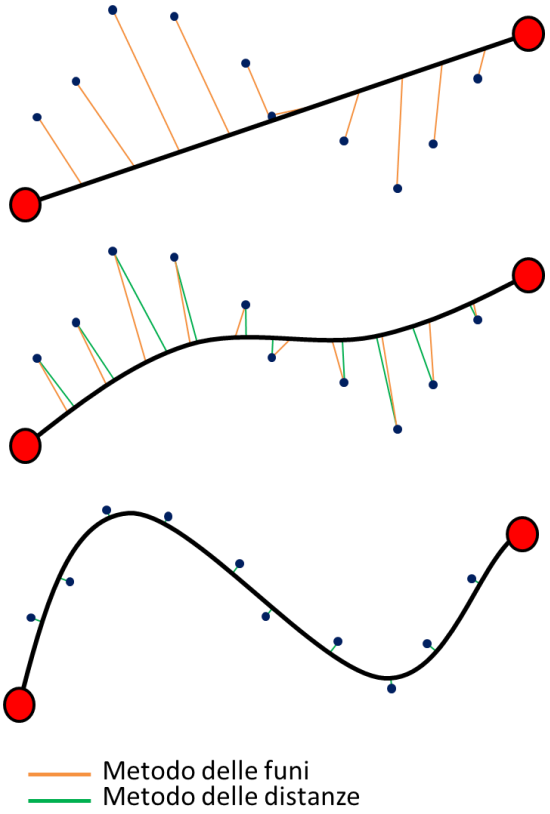


Figura 3.47 – Valutazione dell'errore di approssimazione mediante metodo misto

La possibilità di utilizzo del metodo delle funi deriva dal fatto che i punti che si intendono approssimare nei casi specifici di questa applicazioni sono sempre uniformemente distribuiti e ciò permette di "tirare", proprio come fosse una fune, velocemente la curva in prossimità dei punti stessi.

In Figura 3.48 si mostra l'efficacia del metodo di approssimazione "misto" in un caso di approssimazione avente i punti disposti in modo più articolato rispetto a quelli del caso mostrato in Figura 3.47, in questa situazione, se non si conduce una prima parte della procedura di approssimazione con il metodo delle funi il risultato tenderebbe ad essere quello rappresentato nella parte alta di Figura 3.48; il metodo delle funi permette invece di tenere conto dell'ordine con cui debbono essere fittati i punti e permette di ottenere il risultato mostrato in basso in Figura 3.48.

D'altro canto il solo metodo delle funi produrrebbe un risultato di scarsa accuratezza, per tali motivi si è deciso di impiegare una soluzione di tipo misto.

Inoltre, durante la seconda fase, nella quale avviene il vero e proprio affinamento della curva si tiene di conto del valore di scala di grigio dei singoli voxel definendo una scala di pesi che permette di dare, durante l'approssimazione, maggior importanza ai voxel con elevata fedeltà rispetto alle viste disegnate a mano dal designer (si veda a tal riguardo il paragrafo 3.1.4).

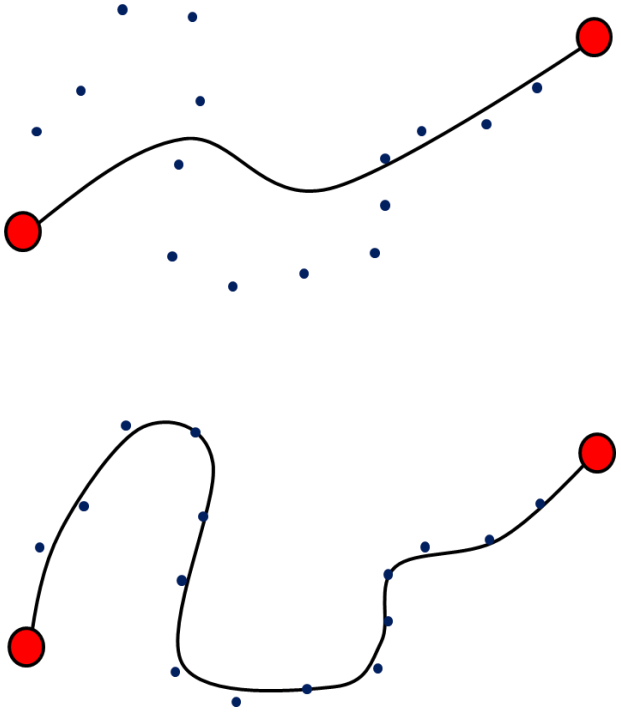


Figura 3.48 – Effetto dell'applicazione del metodo delle funi

Per quanto riguarda la parte elastica dell'errore si è, invece, fatto ricorso alle derivate numeriche prima e seconda della curva in analogia a quanto avviene in molti approcci proposti in letteratura:

$$E_e = E_1 + E_2 = \lambda_1 \int \dot{C}\, ds + \lambda_2 \int \ddot{C}\, ds \qquad (3.8)$$

dove C è la curva ed s la relativa coordinata curvilinea. Nel caso del presente lavoro si è proceduto ad una valutazione della Eq. 3.8 di tipo numerico,

riducendo il calcolo ad una somma di differenze finite; ovvero, sia X la rappresentazione discreta della curva C (discretizzata in n valutazioni) è possibile definire la derivata numerica $\dot{X}$ e la derivata numerica seconda $\ddot{X}$:

$$X = \{X_1;\ X_2;\ X_3;\ \dots\ ;\ X_{n-1};\ X_n\} \tag{3.9}$$

$$\dot{X} = \{X_2 - X_1\ ;\ X_3 - X_2;\ \dots\ ;\ X_n - X_{n-1}\} \tag{3.10}$$

$$\ddot{X} = \{\dot{X}_2 - \dot{X}_1\ ;\ \dot{X}_3 - \dot{X}_2;\ \dots\ ;\ \dot{X}_{n-1} - \dot{X}_{n-2}\} \tag{3.11}$$

Il calcolo dell'errore elastico può quindi essere riformulato come segue:

$$E_e = E_1 + E_2 = \lambda_1 \sum_{n-1}^{i=1} \dot{X}_i + \lambda_2 \sum_{n-2}^{i=1} \ddot{X}_i \tag{3.12}$$

Con λ_1 e λ_2 fattori moltiplicativi che esaltano o meno il grado di levigatezza della curva risultante.

L'assemblaggio della funzione obbiettivo ha quindi richiesto una fase di setup per valutare il corretto valore dei parametri E_p^{th} , λ_1e λ_2.

Il setup è risultato comunque piuttosto veloce permettendo la generazione della soluzione desiderata in tempi rapidi. In Figura 3.49 è possibile vedere come la fase di approssimazione delle curve sia stata condotta con ottimi risultati su di un modello wireframe sufficientemente complesso.

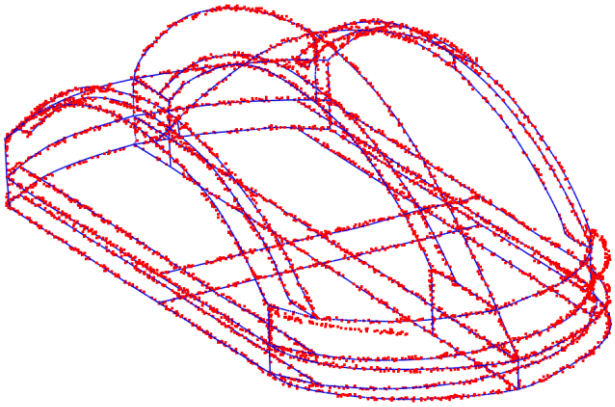


Figura 3.49 – Generazione delle curve finali

Il risultato di questa fase costituisce un primo valido punto di arrivo della metodologia sviluppata qui presentata; il modello ottenuto, infatti, è facilmente esportabile come geometria CAD nei formati di interscambio più diffusi quali DXF, STEP, IGES ecc..

Come descritto nella sezione introduttiva, il processo che porta alla generazione di un modello vettoriale, pur soltanto di tipo wireframe, è un processo che richiede tradizionalmente tempi molto elevati ed è caratterizzato da un'elevata probabilità di errore nell'interpretazione del disegno riprodotto su carta. Il modello fin qui ottenuto costituisce, invece, una rappresentazione fedele ed ottenibile in tempi decisamente più contenuti se confrontati con quelli necessari con metodi tradizionali.

3.5 Generazione delle patch di superficie

L'ultimo tassello della ricostruzione è rappresentato dalla generazione delle superfici. L'insieme di curve determinate nei paragrafi precedenti può essere utilizzato, in quest'ultima fase, allo scopo di definire il set di superfici che meglio descrive il modello rappresentato dai disegni iniziali. Il procedimento di generazione delle superfici può essere condotto manualmente, utilizzando come supporto il modello wireframe vettoriale estratto, oppure può essere automatizzato, ad esempio per mezzo della soluzione qui fornita. Una terza ipotesi di gestione di questa fase può prevedere la generazione automatica di un primo set

di superfici (*superfici di primo tentativo*) sul quale poi andare a lavorare manualmente grazie all'ausilio di sistemi CAD-CAS.

Come anticipato, in questa sede viene presentato un metodo di ricostruzione automatico basato su di una tecnica sviluppata da Bagali e Waggenspack [54]. La tecnica sfrutta le nozioni della teoria dei grafi [55] per determinare un insieme di cicli di spigoli del modello wireframe dal quale estrarre successivamente il modello finale a superfici.

3.5.1 Estrazione dei cicli dal modello wireframe 3D e generazione delle superfici finali

Nel loro lavoro Bagali e Waggenspack utilizzano la teoria dei grafi per descrivere il modello wireframe tridimensionale dell'oggetto che si intende ricostruire. L'impiego delle nozioni sui grafi permette agli autori di trasformare il problema della ricerca delle sequenze di spigoli che definiscono le varie patch di superficie nel problema, ben noto, della determinazione dei cicli di archi all'interno di un grafo. In particolare gli autori sfruttano il ben noto algoritmo di Dijkstra [55] per determinare l'insieme dei cicli.

La tecnica consiste nell'esaminare progressivamente il grafo, partendo da un arco e verificando se e a quali cicli l'arco appartiene. Una volta individuati i cicli legati all'arco in questione si procede alla sua rimozione dal grafo e l'algoritmo prosegue fino all'esaurimento degli archi.

Con riferimento alla Figura 3.50 si può notare come, esaminando l'arco delimitato dai vertici "1" e "2", sia possibile l'identificazione di due diversi cicli. Una volta archiviati questi cicli, si procede alla rimozione dell'arco "1-2" e all'esame dei successivi, "2-3", "1-4", e così via.

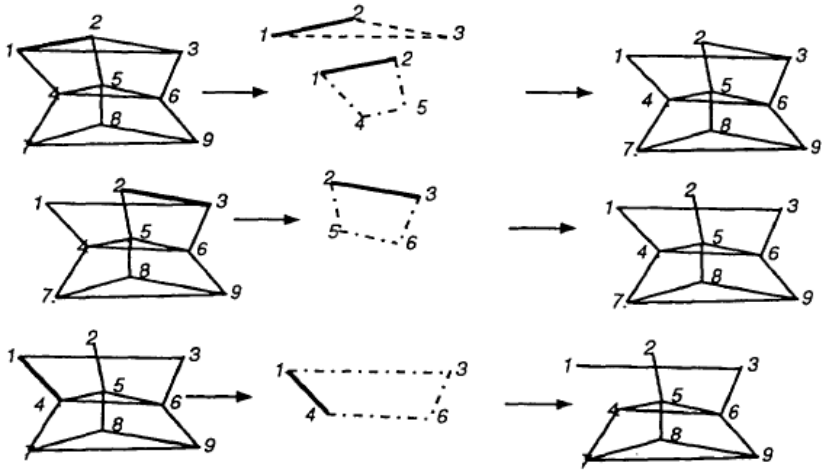


Figura 3.50 – Estrazione del set di cicli

Il risultato di questa procedura consiste in un insieme di cicli di spigoli. La procedura continua quindi con la selezione, all'interno di tale insieme, del set finale di cicli di spigoli che daranno origine alle patch di superfici finali.

E' necessario sottolineare, infatti, come l'insieme di cicli di spigoli individuato non definisca necessariamente una soluzione univoca bensì, più generalmente, rappresenti un sovra insieme della soluzione finale. Questo concetto può essere chiarito con l'aiuto delle immagini se si prende a riferimento la Figura 3.50. Partendo dal modello wireframe analizzato in tale figura è possibile ricostruire una serie di soluzioni finali di cui si forniscono due diversi esempi in Figura 3.51 ed in Figura 3.52.

Nel primo esempio si fornisce la soluzione che prevede che tutte le facce "esterne" del modello wireframe siano convertite in patch di superfice; in tal caso il ciclo $4-5-6$ mostrato in Figura 3.50 non prende parte alla generazione delle patch finali. Diverso il discorso nella soluzione proposta in Figura 3.52 dove il ciclo $4-5-6$ mostrato in Figura 3.50 prende parte alla generazione delle patch finali escludendo per contro il ciclo $1-2-3$.

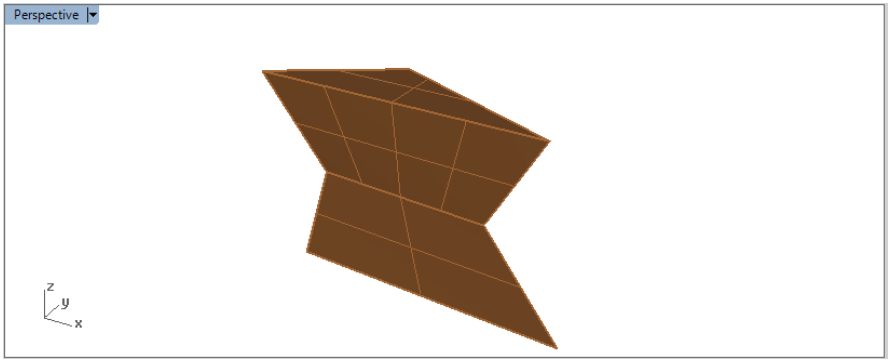


Figura 3.51 – Estrazione del set di cicli

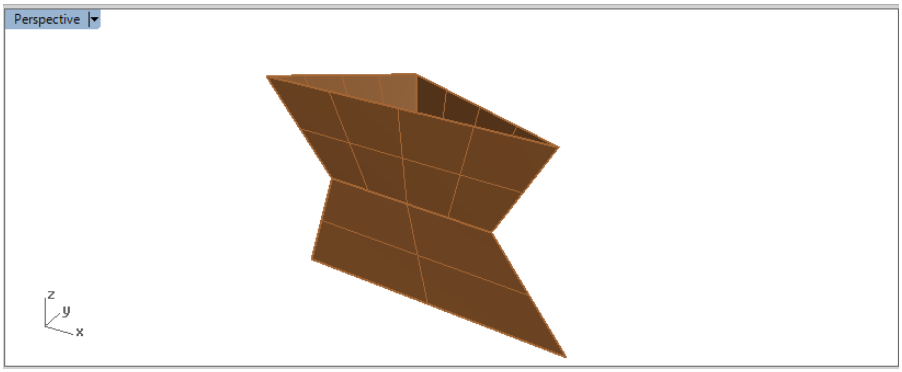


Figura 3.52 – Estrazione del set di cicli

Si fa notare che la non univocità della soluzione per il modello wireframe di Figura 3.50 è legata alla mancanza di vincoli sulla tipologia della soluzione finale; nel caso in cui si fosse vincolati alla generazione di sole geometrie “solide”, l’unica soluzione possibile sarebbe stata quella mostrata in Figura 3.51. E’ noto, tuttavia, che il problema della non univocità della soluzione si presenta anche restringendo il dominio delle soluzioni ai soli oggetti “solidi” (si pensi ad esempio ad un oggetto rappresentato dalle proiezioni di Figura 3.62).

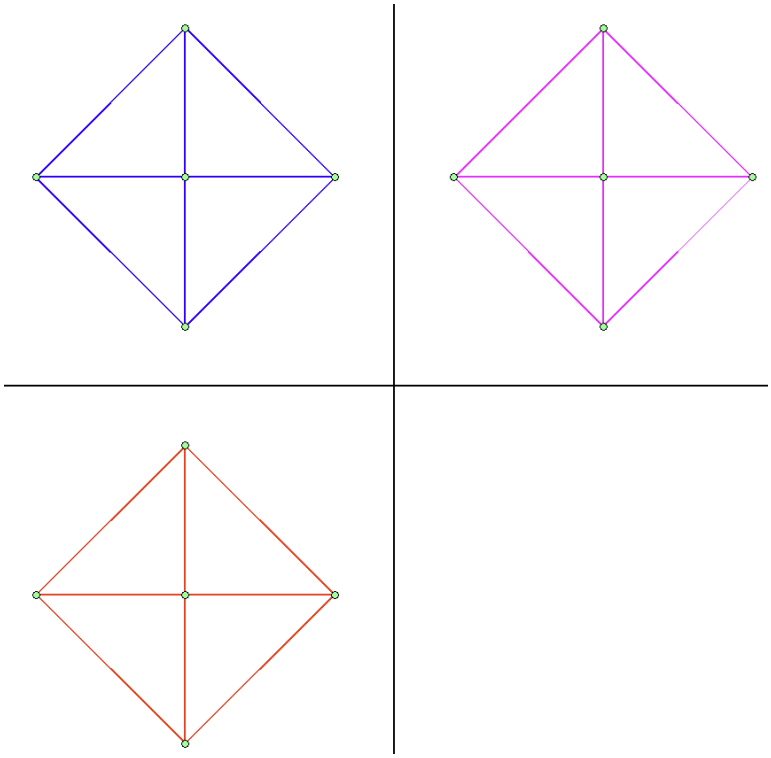


Figura 3.53 – Esempio di proiezioni ortogonali la cui traduzione nello spazio 3D non è univoca

In Figura 3.54 ed in Figura 3.55 si forniscono, per maggior chiarezza, le viste in proiezione ortogonale dei modelli tridimensionali mostrati rispettivamente in Figura 3.51 ed in Figura 3.52.

Proprio in quest'ultima figura si nota l'assenza della patch di superfice superiore generata nell'altra soluzione dal ciclo $1-2-3$ e la presenza della patch "intermedia" generata dal ciclo $4-5-6$.

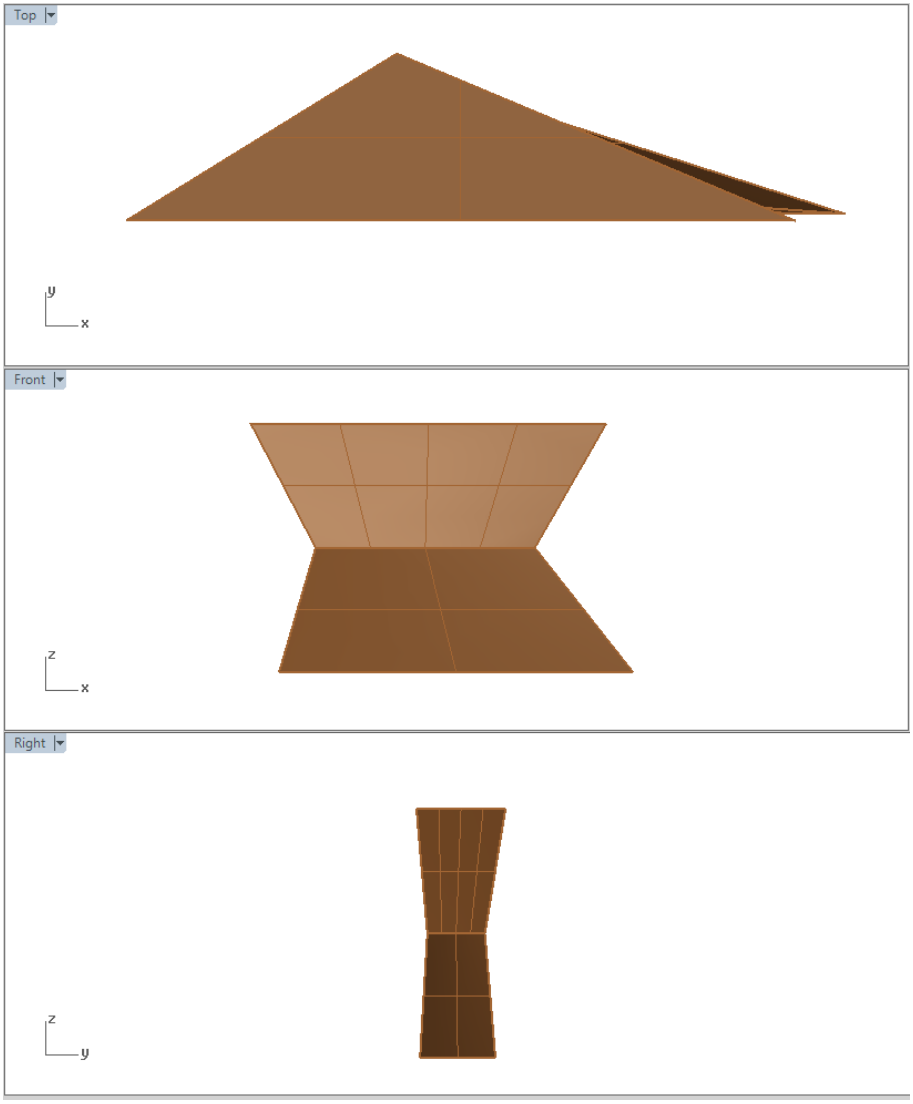


Figura 3.54 – Proiezioni bidimensionali del modello di Figura 3.51

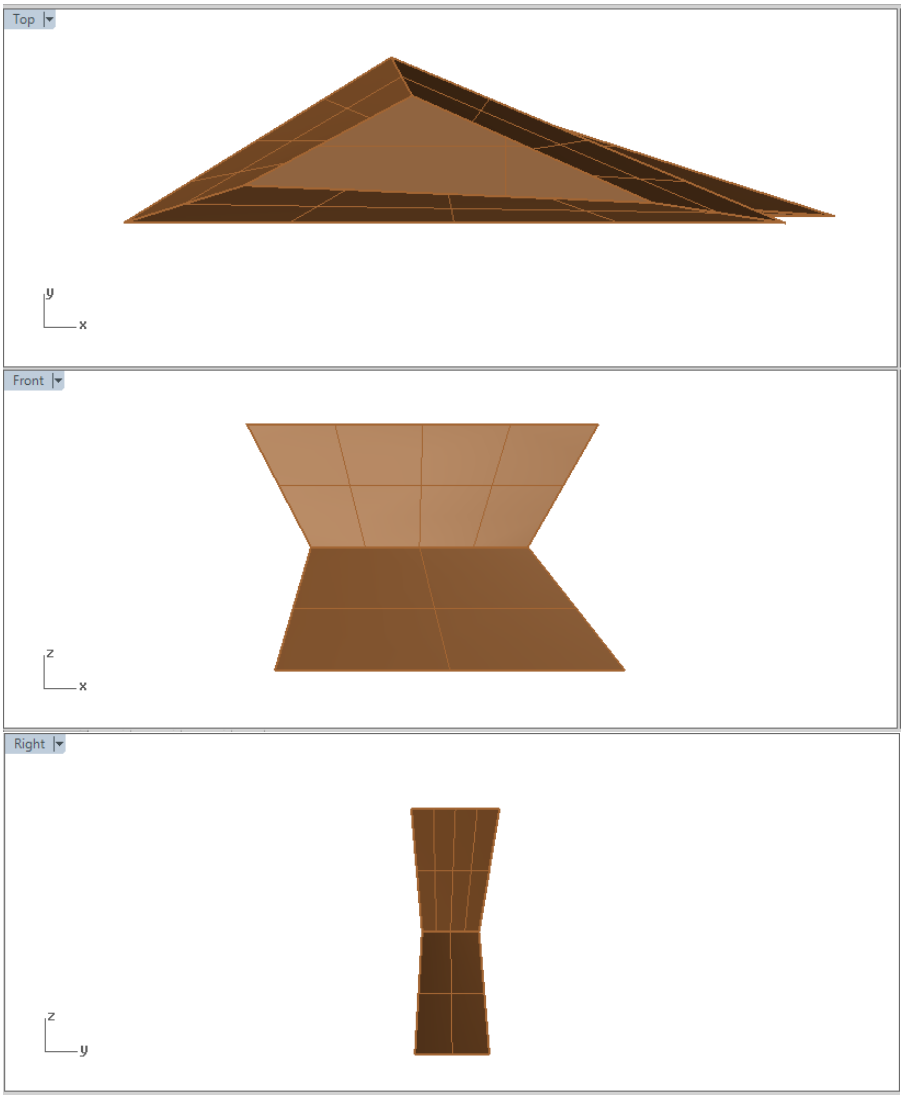
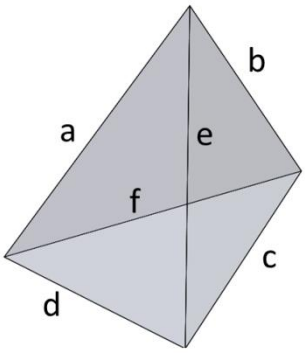


Figura 3.55 – Proiezioni bidimensionali del modello di Figura 3.52

Sulla base di quanto illustrato è evidente il problema di individuare un sottoinsieme dei cicli tale da comporre la soluzione desiderata. Dal punto di vista algoritmico tale processo si traduce nell'individuazione di una base di cicli ottimale.

Una base di cicli si definisce tale in quanto ciascun ciclo in essa non incluso può essere ottenuto da una combinazione di suoi elementi.

Il metodo per generare una base di cicli consiste nello scegliere cicli tra di loro "indipendenti". A tale scopo l'esplorazione delle soluzioni viene eseguita mediante un algoritmo di tipo "greedy" condotto su una base dati (lista dei cicli) avente forma matriciale; in altre parole viene creta una rappresentazione matematica in grado di contenere le informazioni necessarie alla gestione dei cicli individuati sul modello wireframe tridimensionale. In Figura 3.56 si mostra un esempio di tale rappresentazione, il modello grafico di sinistra viene analizzato ed i suoi cicli (f1, f2, f3 ed f4) vengono memorizzati in una matrice che presenta valore unitario in corrispondenza degli spigoli contenuti nel ciclo stesso e 0 altrove.



f1=abf
f2=aed
f3=cdf
f4=bec

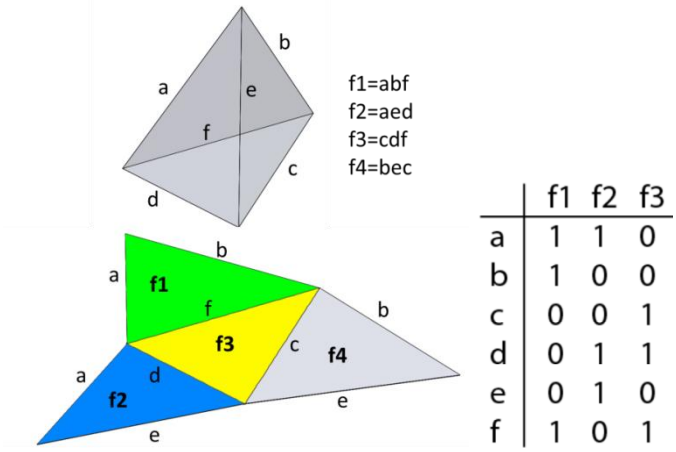
	f1	f2	f3	f4
a	1	1	0	0
b	1	0	0	1
c	0	0	1	1
d	0	1	1	0
e	0	1	0	1
f	1	0	1	0

Figura 3.56 – Rappresentazione matriciale della lista dei cicli

Una volta estratti tutti i possibili cicli si procede alla determinazione di una base, ovvero di un loro sottoinsieme costituito da soli cicli tra loro indipendenti, che rappresenti il set corretto per la generazione delle patch.

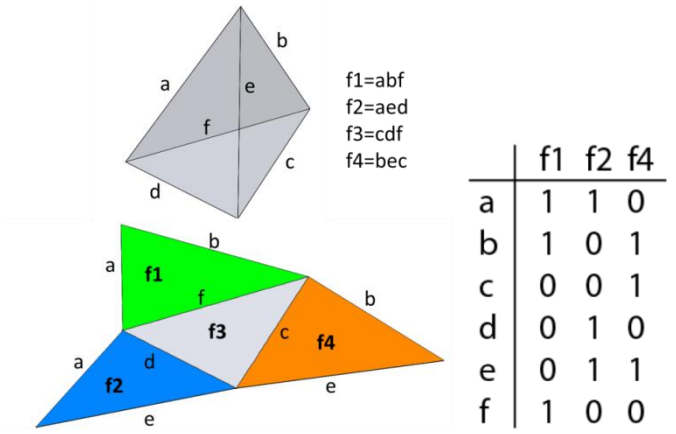
E' necessario a questo punto spiegare il concetto di dipendenza tra cicli; un ciclo risulta dipendente da un insieme di cicli se i suoi spigoli sono inclusi nella lista di spigoli che concorre a formare l'insieme dei cicli stesso. Graficamente il concetto risulta molto semplice: con riferimento alla Figura 3.56 è possibile definire quattro basi alternative, ciascuna delle quali composta da tre cicli poiché il quarto risulterà sempre ottenibile dalla composizione dei primi tre. In Figura 3.57 e Figura 3.58 si mostrano a titolo di esempio due diverse basi, in ciascuna è possibile constatare la dipendenza del ciclo mancante.

Nel primo caso (Figura 3.57) il ciclo f4 è escluso dalla base in quanto risulta dipenderne, infatti lo spigolo c appartiene già alla base tramite il ciclo f3, lo spigolo b tramite il ciclo f1, mentre lo spigolo e fa parte della base in quanto membro di f2; analoghe considerazioni possono essere fatte per l'esempio di Figura 3.58.



	f1	f2	f3
a	1	1	0
b	1	0	0
c	0	0	1
d	0	1	1
e	0	1	0
f	1	0	1

Figura 3.57 – Primo esempio di determinazione di una base di cicli e sua notazione matriciale



	f1	f2	f4
a	1	1	0
b	1	0	1
c	0	0	1
d	0	1	0
e	0	1	1
f	1	0	0

Figura 3.58 – Secondo esempio di determinazione di una base di cicli e sua notazione matriciale

Dal punto di vista implementativo la procedura di discriminazione tra cicli dipendenti e cicli indipendenti richiede l'utilizzo di una branca superiore dell'algebra avanzata conosciuta come teoria dei campi di Galois [56].

Secondo tale teoria è possibile definire un "campo" algebrico GF(2) per il quale valgono le seguenti regole aritmetiche:

- $0 + 0 = 0$
- $1 + 0 = 1$
- $0 + 1 = 1$
- $1 + 1 = 0$

La determinazione dell'indipendenza tra i vari cicli può quindi essere computata come segue:

1- si estrae un ciclo dalla matrice dei cicli (ovvero una colonna);
2- si inserisce tale ciclo nel set dei cicli che andranno a formare una base (ovvero si modifica la matrice dei cicli aggiungendovi la colonna del ciclo estratto);
3- si calcola il rango nel campo GF(2) della matrice;
4- se il rango è pari al numero delle colonne della matrice non vi sono cicli dipendenti e l'algoritmo riprende dal punto 1;
5- se il rango è inferiore al numero delle colonne della matrice significa che il ciclo inserito è dipendente dagli altri; si elimina la sua colonna dalla matrice e si riparte dal punto 1.

La procedura sopra descritta termina allorché tutti i cicli del modello wireframe sono stati esaminati.

Data la procedura di selezione della base di cicli, risulta evidente che l'ordine con cui i cicli vengono esaminati per la loro inclusione o meno nella base gioca un ruolo fondamentale nella determinazione della soluzione finale.

In particolare se l'ordine di esame dei cicli viene costruito secondo una procedura ragionata, dando pesi diversi ai vari cicli e quindi creando una sorta di classifica che veda nella prima posizione il ciclo "migliore" e nell'ultima quello "peggiore", la procedura sopra descritta da luogo ad una base di cicli cosiddetta "ottimale".

Vi sono varie metodologie per dare un punteggio a ciascun ciclo, nel presente lavoro si è privilegiato l'approccio descritto da Abbasinejad et al. in [57]; tale approccio punta all'estrazione della stessa base di cicli che un operatore umano utilizzerebbe per la generazione delle patch di superfice.

A tale proposito, nella determinazione della funzione di assegnazione dei punteggi, vengono utilizzate alcune regole empiriche dettate dall'esperienza maturata, risultano infatti da preferirsi:

- cicli il cui sviluppo sia il più "corto" possibile;
- cicli contraddistinti come *"separating cycle";*
- cicli caratterizzati da piccoli volumi del loro "bounding box" (cicli piccoli e "piatti").

Si introduce quindi anche il concetto di “separating cycle”, ovvero di ciclo la cui rimozione, associata alla rimozione degli spigoli ad esso collegati incrementa il numero di componenti connesse del grafo associato al ciclo ovvero ne provoca la scissione in più componenti distinti.

la fase di generazione di una base ottimale, secondo le regole fin qui illustrate, produce il risultato rappresentato in Figura 3.59 ad eccezione dell’ultimo passaggio, ovvero produce una soluzione “aperta” la cui chiusura deve essere fatta successivamente. Per sua definizione, infatti, una base ottimale non potrà mai contenere il ciclo di “chiusura” in quanto quest’ultimo sarà sempre dipendente dagli altri cicli della base stessa.

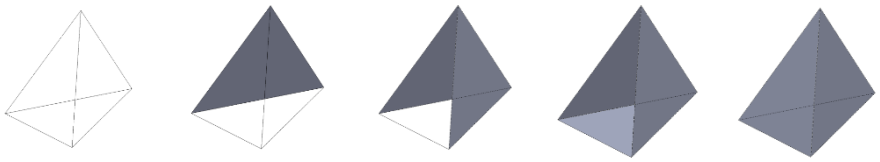


Figura 3.59 – Varie fasi di determinazione del set di cicli finale; l’ultima fase deve essere completata manualmente poiché la base ottimale non prevede per definizione la “chiusura” del modello.

Una volta generata la base di cicli ottimale è possibile procedere alla generazione delle patch di superficie sfruttando un software CAD-CAS (Figura 3.60 e Figura 3.61).

.

Figura 3.60 – Patch finali

Nel caso di geometrie molto complesse, l'identificazione della base di cicli ottimale può risultare alquanto difficile e soggetta ad errori (Figura 3.62). In questi casi si dimostra risolutivo l'intervento dell'operatore che mediante strumenti di modellazione 3D (ad esempio il pacchetto software Rhinoceros®) può lavorare sul modello wireframe apportandovi le opportune semplificazioni. Risulta evidente, infatti, che una geometria semplice, quantomeno composta dai soli spigoli propri del modello wireframe (Figura 3.63), permette la generazione di una base di cicli di miglior qualità dando luogo alla generazione del set di patch più opportuno anche in presenza di modelli complessi.

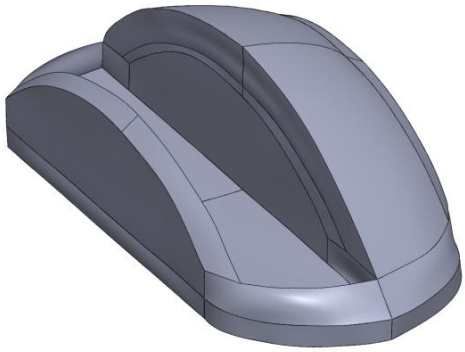


Figura 3.61 – Dettaglio delle patch interne

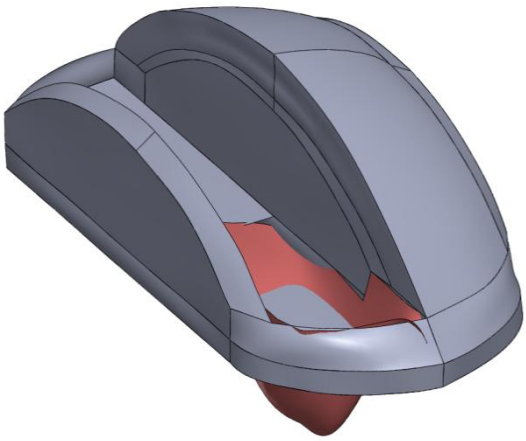


Figura 3.62 – Patch errata (evidenziata in rosso)

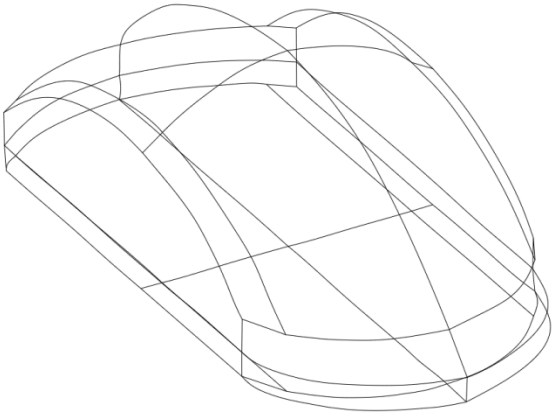


Figura 3.63 – Modello wireframe semplificato

4 Casi studio

La metodologia di ricostruzione sviluppata e proposta in questo lavoro è stata intensivamente collaudata per mezzo di implementazione software prevalentemente condotta in ambiente MatLab®. La parte terminale del lavoro, quella relativa alla creazione delle patch di superfice, è stata implementata e collaudata con l'ausilio del pacchetto software Rhinoceros®.

Nelle catture proposte da Figura 4.1 a Figura 4.8, estratte da un video esplicativo dell'intera procedura, vengono illustrate le varie fasi di ricostruzione eseguite mediante l'implementazione software sopra citata e descritta nel precedente capitolo.

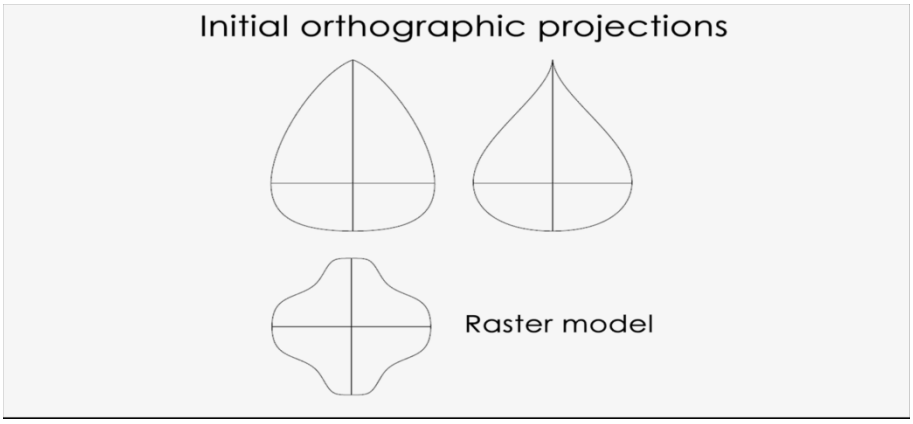


Figura 4.1 – Dato iniziale sotto forma di immagine raster bidimensionale

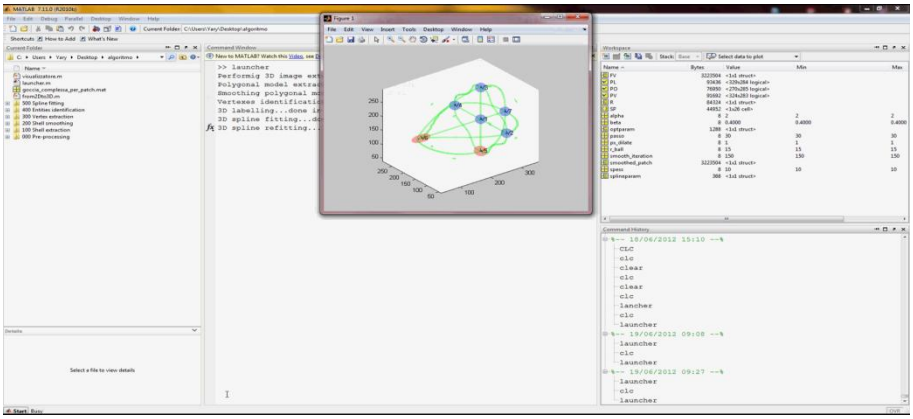


Figura 4.2 – Estrazione del modello wireframe 3D raster (in verde) e dei vertici 3D

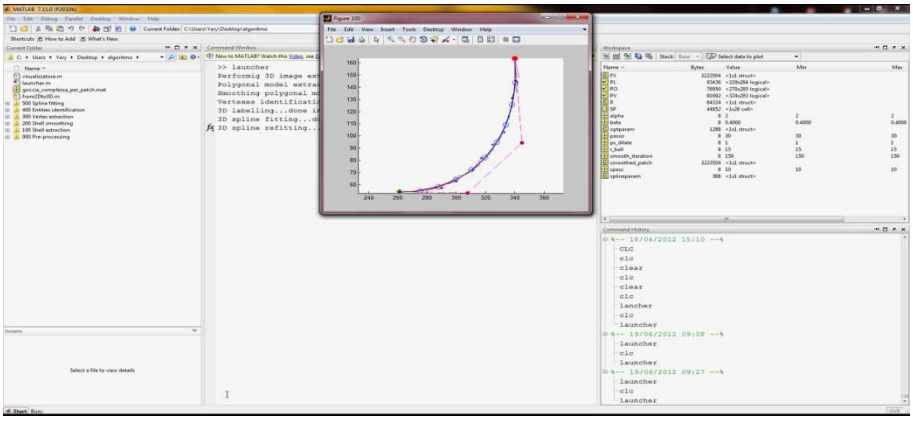


Figura 4.3 – Fitting finale delle curve spline tridimensionali

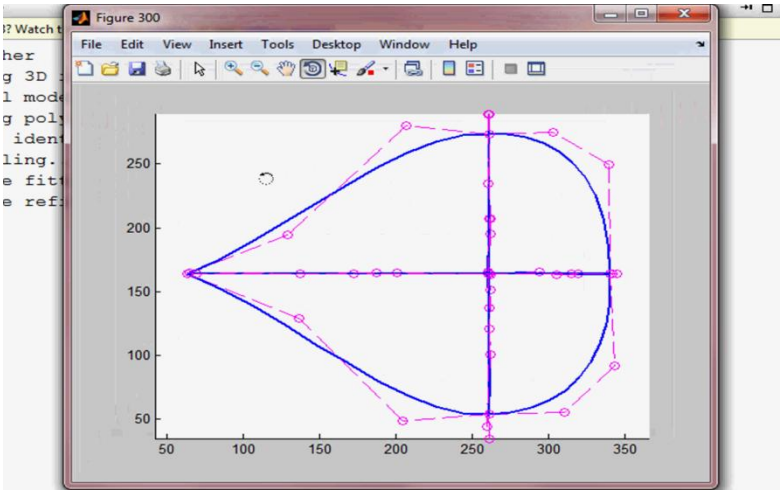


Figura 4.4 – Modello wireframe vettoriale 3D – vista "A"

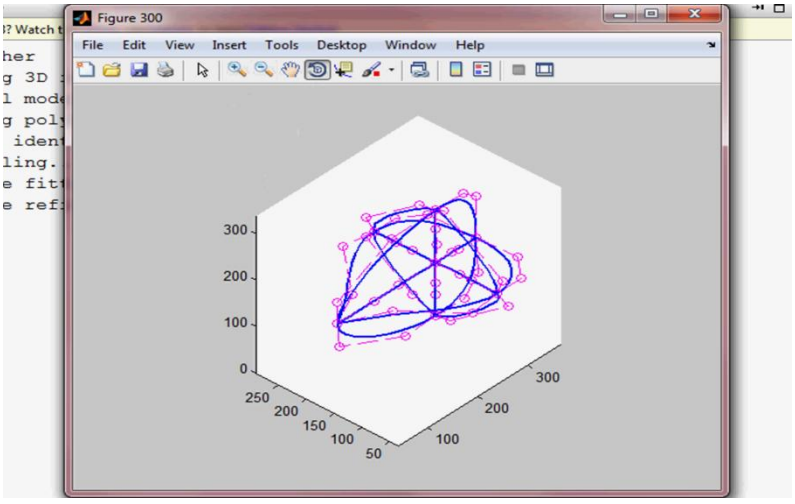


Figura 4.5 – Modello wireframe vettoriale 3D – vista "B"

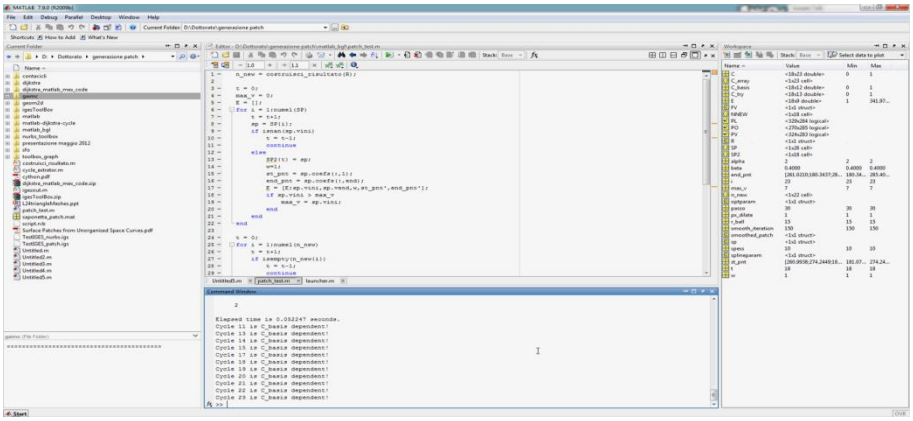


Figura 4.6 – Processo di identificazione della base di cicli ottimale

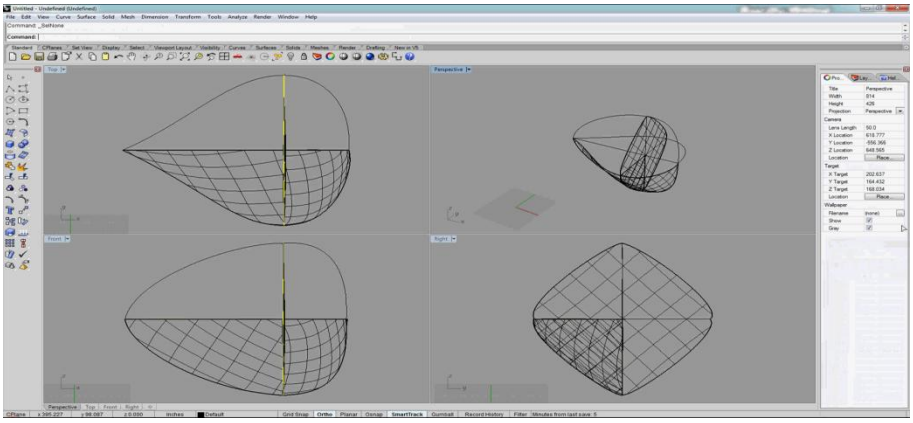


Figura 4.7 – Costruzione delle patch di superfice in ambiente Rhinoceros®

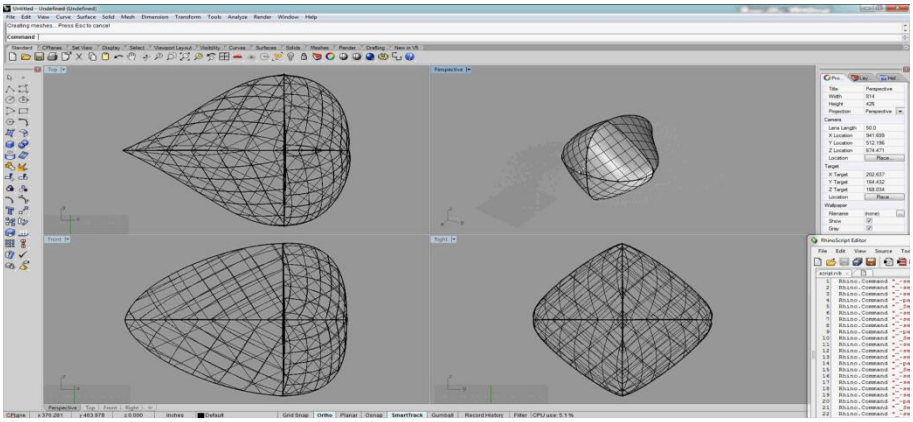


Figura 4.8 – Risultato finale della procedura di riscostruzione

Nella prossima parte di questo capitolo vengono proposti alcuni casi studio impiegati per il collaudo della procedura, in particolare saranno di seguito illustrati cinque casi di ricostruzione a partire da viste ortogonali disegnate a mano libera.

Il primo esempio proposto è quello di un cubo nelle sue tre viste ortogonali (vedi Figura 4.9); in questo esempio, volutamente semplice, si vuole evidenziare un aspetto importante della procedura di ricostruzione descritta in questo lavoro; ossia che l'accuratezza dei bozzetti originali forniti in input al sistema di ricostruzione influisce pesantemente sull'accuratezza di ricostruzione del modello tridimensionale finale e sulla scelta dei parametri di "tuning" della metodologia di ricostruzione.

Nel caso mostrato si nota come occorre iterare la procedura di dilatazione delle immagini bidimensionali più volte prima di ottenere una immagine raster tridimensionale che non abbia lacune in alcuna zona.

In Figura 4.10a è possibile vedere l'immagine 3D ricavata eseguendo soltanto due iterazioni di dilatazione e, come si vede dalla figura, il modello raster presenta molte lacune che ne impediscono la ricostruzione; in Figura 4.10b è possibile notare come, passando a sei iterazioni di dilatazione, si ottenga un modello raster di qualità inferiore (in quanto caratterizzato da una rappresentazione wireframe dove gli spessori sono rilevanti) ma indubbiamente completo e capace di essere processato dalla rimanente parte della procedura. In Figura 4.10c si mostra il modello wireframe ricostruito ed infine in Figura 4.10d è possibile osservare il modello a patch di superfici finale.

A titolo di riflessione è possibile affermare che in questo caso la presenza di geometrie ben distinte e di entità con angoli di incidenza molto pronunciati (per definizione essendo un cubo tutti gli angoli sono prossimi all'angolo retto)

permette una certa libertà nella gestione della fase di dilatazione. Nel prossimo esempio verrà affrontato proprio questo tipo di problema e sarà possibile capire che, in mancanza delle condizioni sopracitate, un abuso della procedura di dilatazione porta a un rapido degrado del modello che si intende ricostruire.

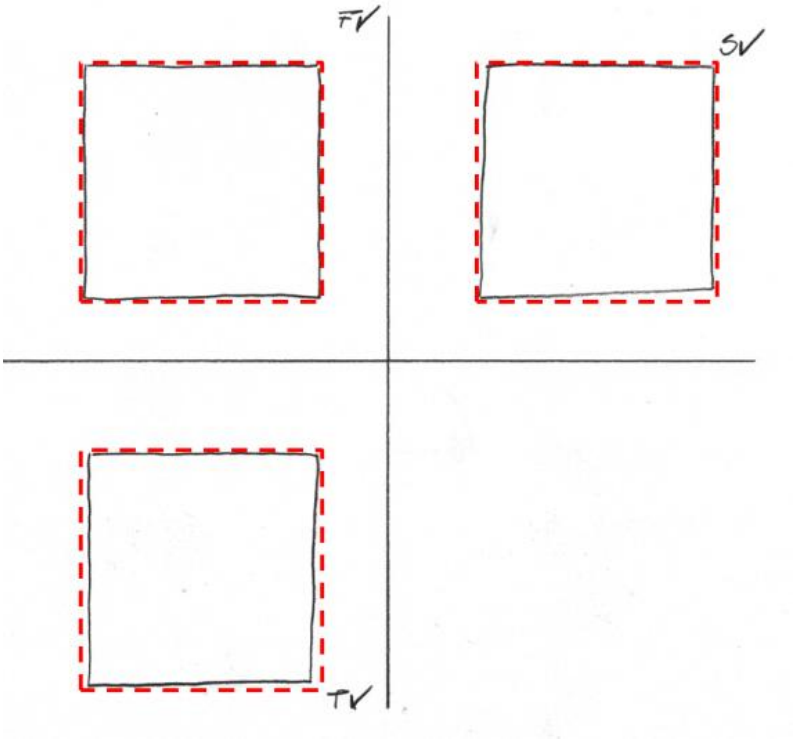


Figura 4.9 – Caso studio "A" – proiezioni ortogonali e bounding box

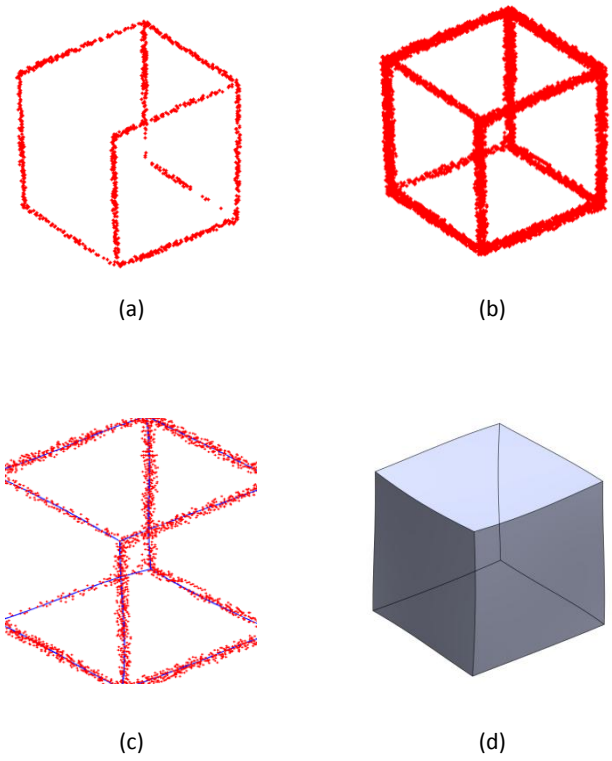


Figura 4.10 – Caso studio "A" – Immagine 3D e Modello CAD 3D

Il secondo caso tratta un modello in cui gli angoli di incidenza tra le varie entità non sono ben marcati, in particolare nella Side View mostrata in Figura 4.11. In questo caso si nota come la topologia del modello wireframe, proposto in Figura 4.12 non sia corretta in quanto la traduzione dei bozzetti ha generato una entità topologica non corretta (la curva gialla visibile in Figura 4.12).

Questo errore si ripercuote sulla soluzione finale, è infatti possibile vedere in Figura 4.13 un leggero "difetto" del modello proprio in prossimità della zona nella quale si trova lo spigolo incriminato.

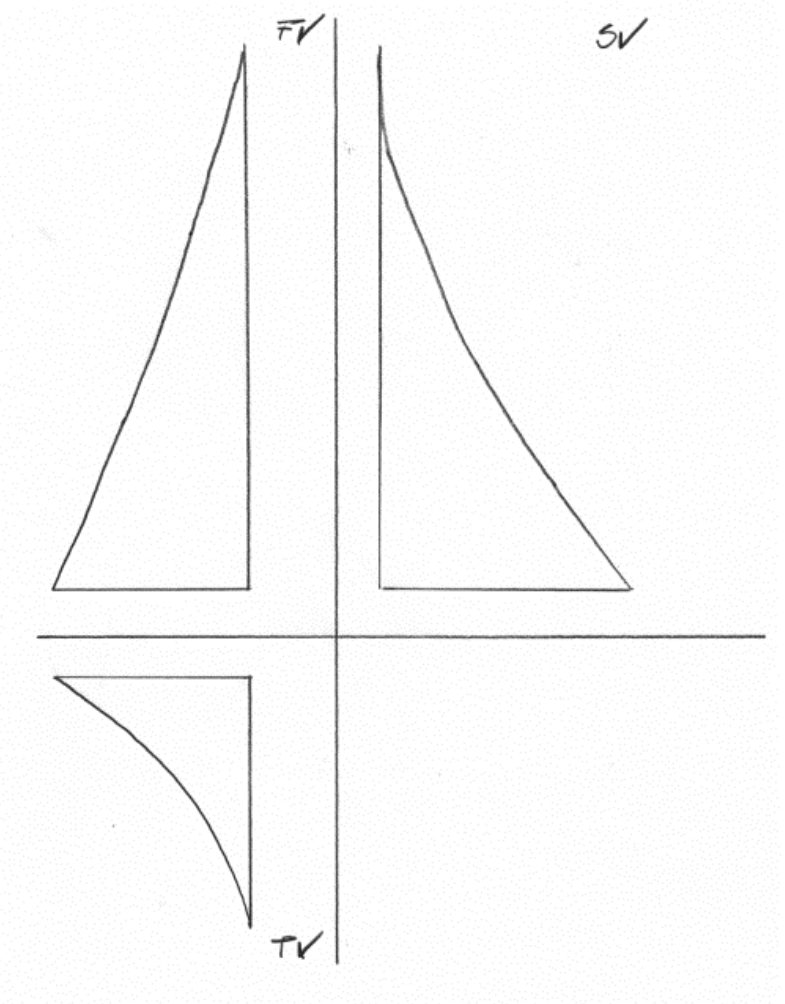


Figura 4.11 – Caso studio "B" – proiezioni ortogonali

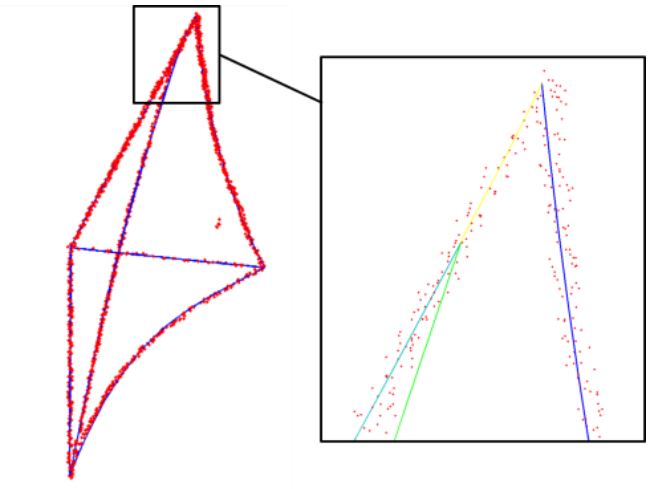


Figura 4.12 – Caso studio "B" – Wireframe 3D

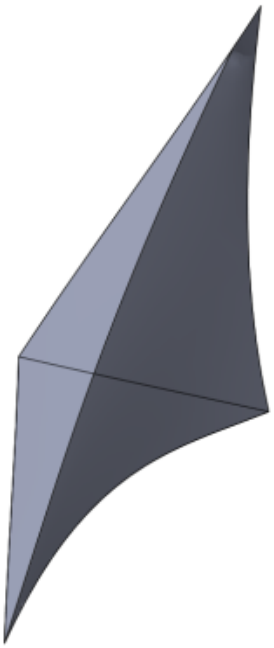


Figura 4.13 – Caso studio "B" – Modello CAD 3D

Nel terzo caso presentato si vede come sia possibile superare questi inconvenienti ponendo una maggior attenzione nella realizzazione dei bozzetti di partenza. Nonostante la geometria in Figura 4.14 – risulti più complessa rispetto alla precedente, la procedura di ricostruzione viene eseguita senza alcun tipo di problema e produce il risultato corretto visibile in Figura 4.19.

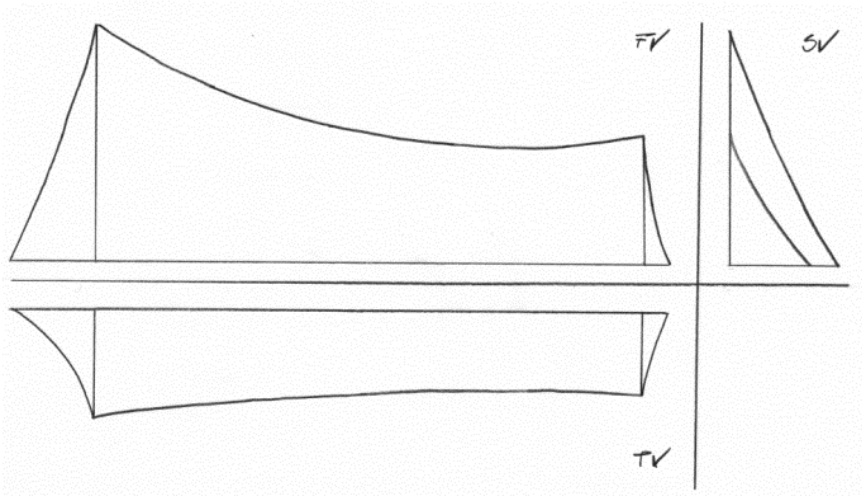


Figura 4.14 – Caso studio "C" – proiezioni ortogonali

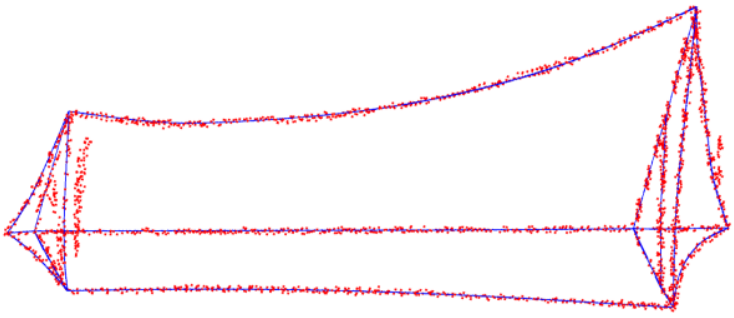


Figura 4.15 – Caso studio "C" – Wireframe 3D

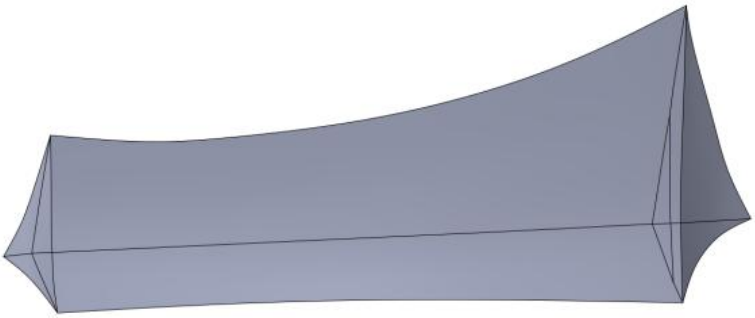


Figura 4.16 – Caso studio "C" – Modello CAD 3D

Il caso studio "D", presentato in Figura 4.17, mostra una struttura "a goccia" nella quale si evidenzia una certa variabilità nella curvatura delle linee di stile, soprattutto riguardo a quelle visibili nel piano orizzontale (TV). La Figura 4.19 mostra le superfici generate al termine della procedura di ricostruzione. Anche in questo caso la procedura ha avuto successo grazie alla bontà di definizione delle zone di intersezione tra le varie curve (visibile sia in proiezione nella Figura 4.17 che in vista 3D nel modello wireframe di Figura 4.18).

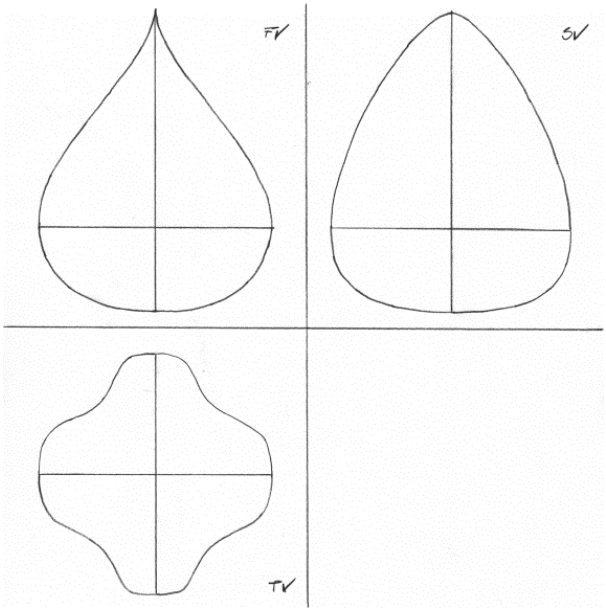


Figura 4.17 – Caso studio "D" – Bozzetti 2D

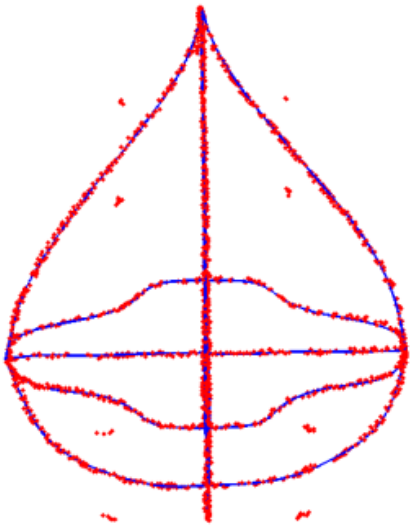


Figura 4.18 – Caso studio "D" – Wireframe 3D

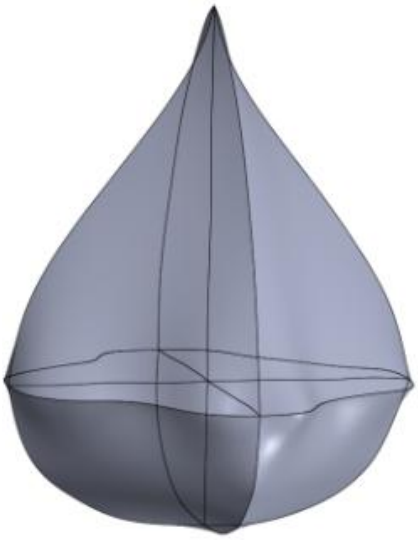


Figura 4.19 – Caso studio "D" – Modello CAD 3D

L'ultimo caso presentato è il caso studio "E" le cui proiezioni di partenza sono rappresentate in Figura 4.20. La peculiarità di questo caso riguarda l'elevato numero di entità topologiche ricostruite e visibili nel modello wireframe di Figura 4.21. Come si nota dal modello wireframe sono infatti presenti numerose zone del disegno in cui l'intersezione delle tre viste ha generato artefatti (gruppi di voxel isolati) non appartenenti al modello 3D che la procedura deve fornire come risultato. Nonostante questi "disturbi" il flusso di ricostruzione procede senza intoppi e fornisce il risultato corretto e visibile in Figura 4.22.

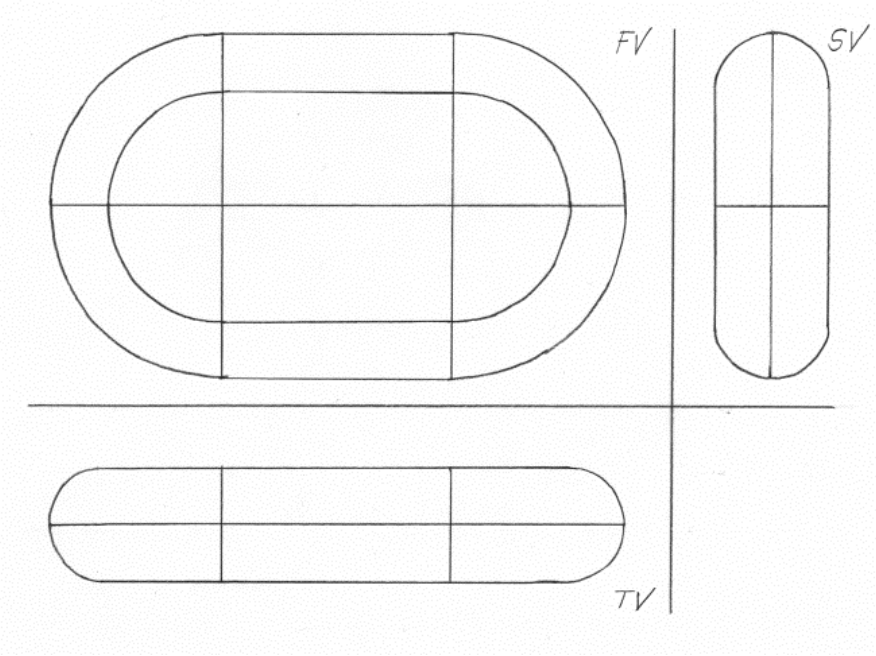


Figura 4.20 – Caso studio "E" – Bozzetti 2D

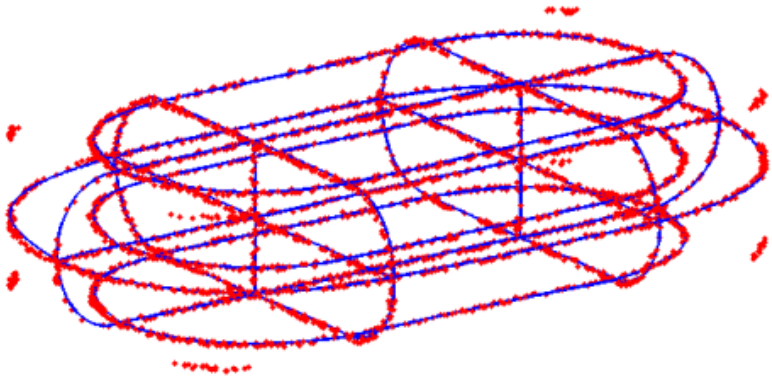


Figura 4.21 – Caso studio "E" – Wireframe 3D

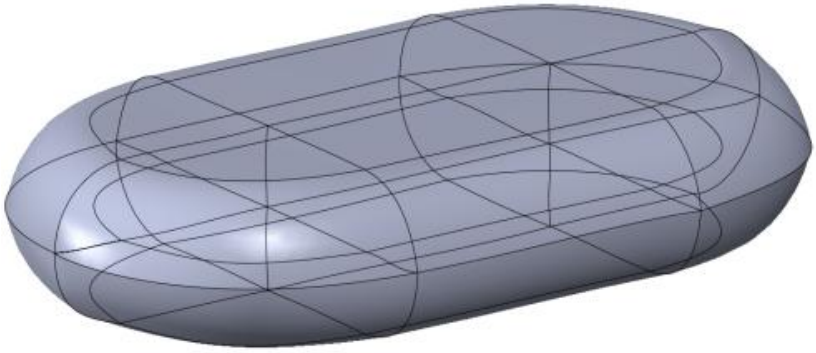


Figura 4.22 – Caso studio "E" – Modello CAD 3D

Di seguito si presenta un confronto dei tempi di esecuzione dell'algoritmo di ricostruzione impiegato per la realizzazione dei casi studio appena presentati ed i tempi necessari alla ricostruzione con tecnica tradizionale. Per meglio definire i termini di paragone delle tempistiche si specifica che i tempi non tengono conto della taratura dei parametri dell'algoritmo di ricostruzione.

Per quanto concerne la ricostruzione con tecnica tradizionale si è adottato il seguente protocollo:

FASE A. acquisizione delle tre proiezioni tramite scanner piano;
FASE B. separazione delle viste con software generico per creazione di tre immagini distinte (Figura 4.23);
FASE C. importazione e disposizione delle immagini in ambiente CAD 3D (Figura 4.24);
FASE D. ricostruzione del modello wireframe 3D (Figura 4.25);
FASE E. generazione delle patch di superfice in ambiente CAD 3D (Figura 4.26).

La tabella sottostante contiene i valori emersi dalla comparazione delle tecniche di ricostruzione condotte sui casi studio "D" ed "E" (in tabella rispettivamente "CSD" e "CSE"). Tutti i valori di tempo sono espressi in secondi.

	Tecnica Tradizionale		*Ricostruzione automatica*	
	CSD	*CSE*	*CSD*	*CSE*
FASE A (s)	67	75	67	75
FASE B (s)	178	173	320	745
FASE C (s)	287	325		
FASE D (s)	880	2960		
FASE E (s)	105	235		
Totale (s)	**1517**	**3768**	**387**	**820**
Incremento di produttività			**4x**	**4,5x**

Come considerazione generale sui dati ottenuti dai suddetti confronti si evidenzia come la procedura di ricostruzione automatica sia stata implementata in ambiente Matlab® al solo scopo di verificare qualitativamente la bontà dei risultati; implementazioni volte alla ingegnerizzazione degli algoritmi utilizzati e/o del codice

Matlab® permetterebbero un miglioramento delle prestazioni quantificabile nella riduzione dei tempi di calcolo di almeno un ordine di grandezza.

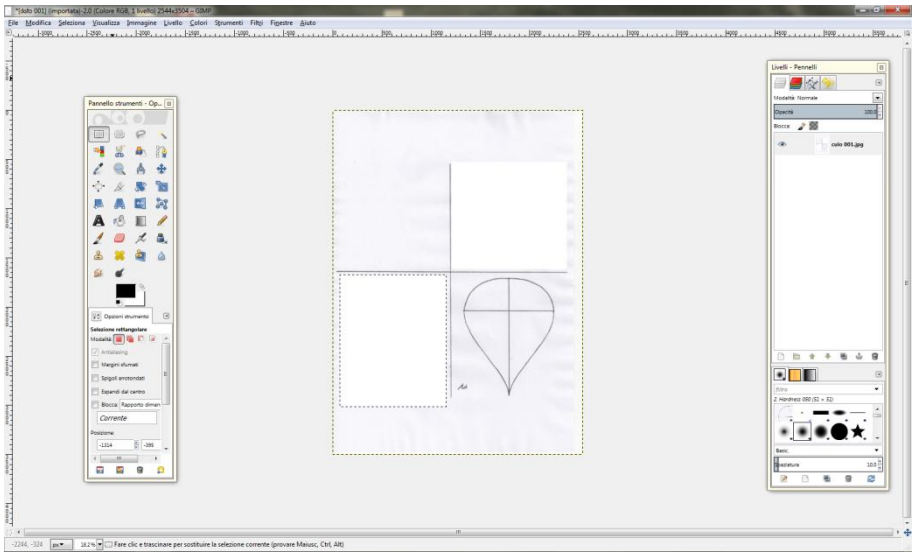


Figura 4.23 – Elaborazione dell'immagine acquisita in ambiente GIMP

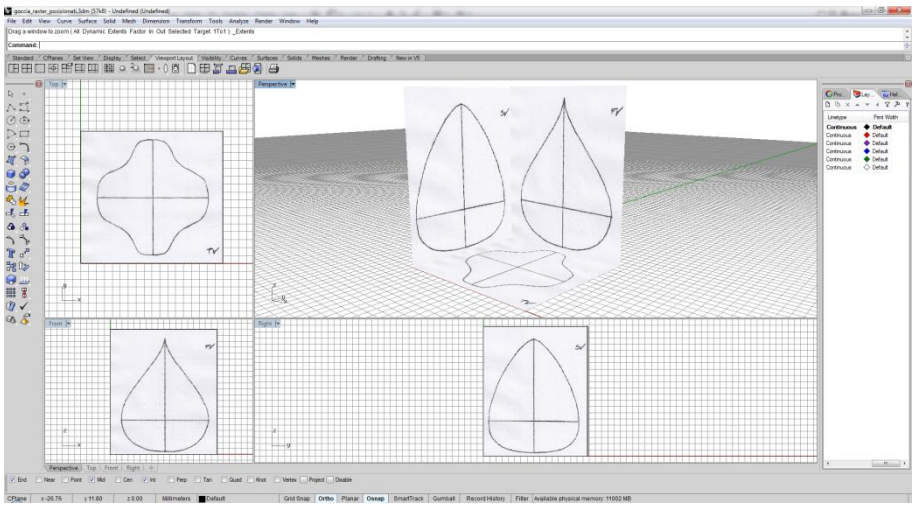


Figura 4.24 – Posizionamento delle immagini sui piani coordinati in ambiente Rhinoceros®

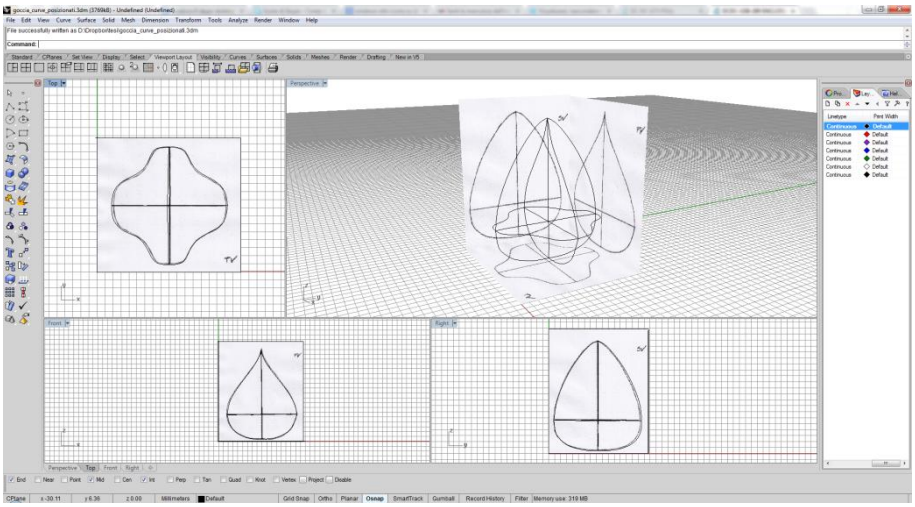


Figura 4.25 – Generazione del modello wireframe 3D in ambiente Rhinoceros®

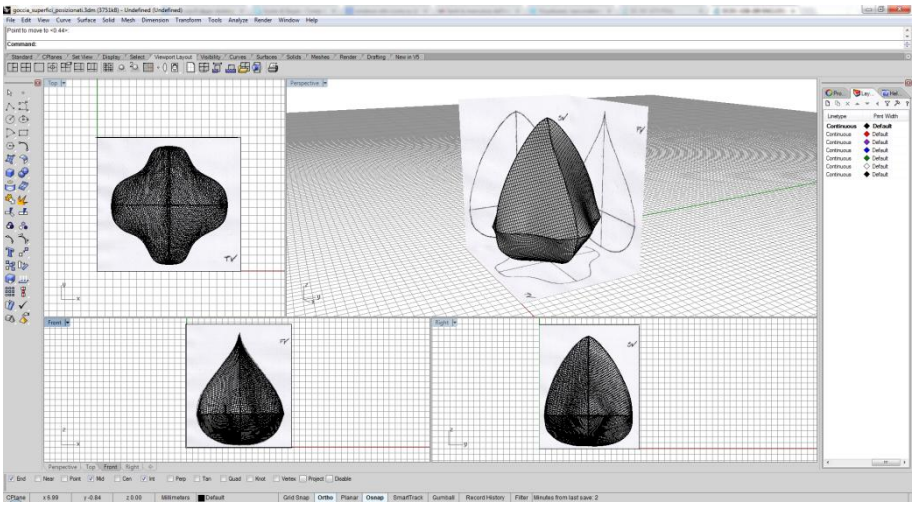


Figura 4.26 – Generazione del modello 3D a superfici in ambiente Rhinoceros®

5 Conclusioni

Negli ultimi decenni la progettazione ha ricevuto un importante contributo dall'utilizzo del computer, con notevole impatto in termini di tempi, costi e affidabilità. Se, nella normalità dei casi, il computer è sicuramente il punto di partenza del processo di sviluppo di nuovi prodotti, in molti casi ciò non può ancora avvenire, oppure avviene ma con procedure lente e difficoltose. Si pensi ai casi in cui l'elemento stilistico assume un ruolo primario, per esempio nel caso di prodotti di design (nell'accezione italiana del termine); convenzionalmente, lo schizzo a mano libera continua ad essere uno dei metodi preferiti, in particolar modo nelle prime fasi del design stilistico.

Creare un modello CAD tridimensionale a partire da questi schizzi è, tuttavia, un processo tanto laborioso con gli strumenti attualmente disponibili quanto necessario per restare al tempo con le moderne tecniche di progettazione e di produzione. Analoghe problematiche persistono anche nelle fasi più avanzate del design ovvero quando il bozzetto ha assunto una sua ben definita geometria e mostra forti analogie con i disegni tecnici.

L'obiettivo della presente attività di tesi è stato, quindi, quello di sopperire a questa lacuna, superando l'attuale stato dell'arte e della tecnica. In particolare, posando lo sguardo su campi di ricerca affini, è stata sviluppata una metodologia innovativa per la traduzione automatica di disegni tecnici di stile in geometrie CAD tridimensionali. Sono stati, infatti, esaminati una serie di lavori appartenenti al filone di ricerca della *3D reconstruction* che costituivano, all'inizio del presente percorso di dottorato, lo stato dell'arte più recente delle tecniche di ricostruzione 3D da dati bidimensionali.

Parallelamente, il dominio di ricerca è stato esteso al campo della *computer vision* con particolare riguardo alla gestione ed alla manipolazione delle immagini tridimensionali.

Sono state quindi progettate, affinate ed infine implementate, prevalentemente in ambiente di sviluppo Matlab®, una serie di procedure che

compongono la metodologia di ricostruzione presentata all'interno del Capitolo 3 della presenti tesi di Dottorato.

Il lavoro è stato organizzato secondo un flusso logico che ha portato ad affrontare in modo sequenziale ed ordinato ciascuna fase della ricostruzione proponendo, dove possibile, soluzioni alternative. Ne è risultata una metodologia caratterizzata da una buona robustezza e da un elevato grado di automazione che limita al minimo l'interazione con l'utente durante il procedimento di ricostruzione (il Capitolo 4 presenta soltanto alcuni tra i molti casi studio affrontati nel corso dello svolgimento dell'attività del Dottorato). Per ciascun problema affrontato sono state elaborate, implementate e validate più soluzioni, sia ricorrendo allo sviluppo di idee originali che traendo spunto da quanto già proposto in letteratura.

La prima fase del processo di ricostruzione permette, partendo dal dato raster 2D, di ottenere con estrema rapidità e fedeltà una versione raster tridimensionale dello schizzo di partenza. Le successive procedure di identificazione della topologia e di generazione delle entità vettoriali, in linea di principio condotte in modo del tutto automatico, costituiscono un'accelerazione importante rispetto al processo di ricostruzione convenzionale; anche nei casi più complessi, nei quali può rendersi necessaria l'interazione con l'utente il procedimento di ricostruzione si rivela assai più veloce rispetto al processo convenzionale.

Una delle maggiori criticità che si sono evidenziate nel corso dello sviluppo della metodologia presentata, riguarda senza dubbio la forte dipendenza tra la qualità del dato in ingresso (bozzetto raster) ed il risultato finale della ricostruzione; risulta infatti di semplice comprensione come un dato raster bidimensionale non sufficientemente "definito" pregiudichi fortemente il corretto processo di ricostruzione come evidenziato Capitolo 4 dal caso studio "B".

Dall'analisi dei molti casi studio e sulla base dell'esperienza maturata durante il lavoro è possibile concludere che la soluzione a tale criticità non può prescindere da una migliore qualità nello schizzo manuale di partenza; tale requisito può essere considerato come uno dei pochi vincoli posti al designer affinché si renda possibile l'utilizzo di una procedura di ricostruzione come quella presentata nel presente lavoro.

Inoltre, si evidenzia come alcuni aspetti trattati nel periodo del presente Dottorato hanno mostrato delle problematiche verosimilmente destinate ad essere risolte con il passare del tempo. Da un punto di vista concettuale, ad esempio, alcune tecniche come quelle relative alla scheletonizzazione tridimensionale hanno dimostrato grossi limiti riconducibili alla loro scarsa maturità; da un punto di vista pratico la necessità di lavorare con immagini tridimensionali ha evidenziato limiti sia dal punto di vista della complessità computazionale degli algoritmi sia dal punto di vista della enorme mole di dati da gestire.

In ragione anche di tali motivi, la metodologia sviluppata e presentata all'interno dell'attività di tesi, non è e non vuole essere un punto di arrivo ma

vorrebbe rappresentare un punto di partenza per lo sviluppo e/o l'affinamento di ulteriori strumenti per la ricostruzione.

Allo scopo di favorire quanto appena asserito la traccia principale della presente attività di Dottorato e molte delle tecniche sviluppate nel corso del presente lavoro sono state oggetto di pubblicazioni scientifiche [58–65].

Sia nel campo della ricerca accademica che in quello dello sviluppo industriale potranno quindi essere tratti spunti dalla metodologia presentata; è possibile, infatti, immaginare almeno due diversi scenari evolutivi, il primo volto ad indagare ulteriormente ed estendere le potenzialità delle varie parti della metodologia oggetto della presente tesi, l'altro mirato alla industrializzazione e quindi alla diffusione su larga scala delle tecniche in essa proposte.

Da un punto di vista più strettamente scientifico si possono sviluppare ulteriori tecniche che arricchirebbero gli strumenti forniti dalla presente metodologia; alcuni esempi potrebbero riguardare integrazioni che aumentino le capacità della metodologia presentata in termini di:

1- identificazione delle relazioni di tipo geometrico come tangenze, ortogonalità, simmetrie ecc.;
2- acquisizione di informazioni da un numero di viste superiore a tre;
3- acquisizione di informazioni da rappresentazioni di tipo assonometrico;
4- modellazione delle patch di superfice guidata dalle informazioni contenute nelle viste bidimensionali (ad esempio tramite tecniche proprie dello Shape from Shading [66] o dello Shape from Texture).

Sviluppi di tipo industriale potrebbero invece riguardare l'ingegnerizzazione delle tecniche proposte e quindi la loro integrazione con pacchetti software CAD diffusi in ambito industriale; tale scenario darebbe maggiore continuità al lavoro di ricostruzione offrendo la possibilità all'utente finale di usufruire contemporaneamente degli strumenti sviluppati *ad-hoc* per la ricostruzione e dei tipici *tool* messi a disposizione dai moderni software CAD tridimensionali.

È infine importante sottolineare come le varie tecniche illustrate nei capitoli precedenti costituiscono una serie di strumenti efficaci, comunque migliorabili e/o intercambiabili con altri che dimostrino capacità superiori. Si vuole invece rimarcare come il contributo di maggior rilievo apportato dalla presente attività di tesi consiste nell'aver proposto una metodologia nuova, totalmente originale, per la ricostruzione di modelli 3D computazionali di tipo free-form a partire da disegni di stile creati a mano libera. Le varie tecniche proposte non sono altro che il set di strumenti scelti, sviluppati e modellati allo scopo di accreditare la metodologia ideata.

Elenco delle Figure

Figura 1.1 – Descrizione grafica del processo di ricostruzione 3D di un disegno tecnico di stile su carta 18

Figura 1.2 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software SolidWorks®: posizionamento relativo delle viste bidimensionali nello spazio di lavoro 19

Figura 1.3 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software SolidWorks®: creazione del modello 3D mediante utilizzo delle informazioni contenute nelle viste bidimensionali. 20

Figura 1.4 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software SolidWorks®: modello tridimensionale risultante dal lavoro di traduzione delle viste bidimensionali 21

Figura 1.5 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software Inventor®: viste iniziali tipiche del disegno 2D. 22

Figura 1.6 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software Inventor®: posizionamento relativo delle viste bidimensionali nello spazio di lavoro. 23

Figura 1.7 – Descrizione grafica del processo di ricostruzione 3D implementato nel pacchetto software Inventor®: modello tridimensionale risultante dal lavoro di traduzione delle viste bidimensionali. 24

Figura 1.8 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: viste iniziali. 25

Figura 1.9 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: posizionamento relativo delle viste iniziali nello spazio di lavoro. 25

Figura 1.10 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: tracciamento manuale delle curve di stile sulle proiezioni bidimensionali di partenza. 26

Figura 1.11 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: definizione delle patch di superfice giacenti sulle curve di stile estratte al passaggio precedente. 26

Figura 1.12 – Descrizione grafica del processo di ricostruzione 3D effettuato mediante il pacchetto software CATIA®: modello tridimensionale ottenuto al termine della ricostruzione. 27

Figura 1.13 – Disegno di stile in stato avanzato di elaborazione 28

Figura 2.1 – Cicli planari 35

Figura 2.2 –Bridge all'interno di cicli planari 36

Figura 2.3 – Due virtual face correttamente giuntate 38

Figura 2.4 – Intersezioni irregolari di primo tipo 38

Figura 2.5 – Intersezioni irregolari di secondo tipo 39

Figura 2.6 – Proiezioni e wire frame di un oggetto contenente piani di simmetria. 40

Figura 2.7 – Soluzioni legate alle proiezioni di figura 2.6 41

Figura 2.8 – Particolare privo di piani di simmetria 41

Figura 2.9 – Particolare ricostruito con la tecnica elaborata da Liu et al. 45

Figura 2.10 – Classe dei segmenti nel lavoro di Gong et al. 46

Figura 2.11 – Classe delle curve coniche nel lavoro di Gong et al. 47

Figura 2.12 – DFS, ordine di esplorazione dei nodi 48

Figura 2.13 – Albero delle configurazioni per la vista frontale 48

Figura 2.14 – Esplorazione dell'albero delle configurazioni 50

Figura 2.15 – Flusso del lavoro nella metodologia di Kara ed Eramo 51

Figura 2.16 – Allineamento di template e immagine 51

Figura 2.17 – Patch di primo tentativo 52

Figura 2.18 – Modellazione delle patch in post-produzione 52

Figura 2.19 – Applicazione della metodologia di Kara ed Eramo al design in campo automobilistico 55

Figura 3.1 – Bozzetto di stile disegnato su supporto cartaceo 58

Figura 3.2 – Interpretazione del risultato della scansione 59

Figura 3.3 – Bounding box delle tre proiezioni 60

Figura 3.4 – Dettaglio dell'operazione di dilatazione ed effetto dei pesi; dall'alto in basso a sinistra si mostrano in successione: immagine originale, primo step (grigio 75%), secondo step (grigio 50%) e terzo step (grigio 25%). A destra le versioni a colori per maggior facilità di interpretazione. 62

Figura 3.5 – Modello wireframe ottenuto con $Ns = 0$ 64

Figura 3.6 – Modello wireframe ottenuto con $Ns = 2$ 64

Figura 3.7 – Metodo delle distanze per il calcolo delle zone di intersezione 66

Figura 3.8 – Metodo delle distanze – casi critici 66

Figura 3.9 – Metodo del gradiente per il calcolo delle zone di intersezione 67

Figura 3.10 – Metodo del gradiente – casi critici 68

Figura 3.11 – Rappresentazione grafica dei k-vicinati con $k = 6$, $k = 18$ e $k = 26$ 69

Figura 3.12 – Direzioni individuabili nel vicinato $N26$ 70

Figura 3.13 – Ordinamento dei valori di stima dello spessore riferiti al singolo voxel 71

Figura 3.14 – Esempio di individuazione della direzione principale; Da indica la direzione principale reale, $Dest$ quella stimata compiendo valutazioni sulle sole 13 direzioni esaminate. 72

Figura 3.15 – Gestione dei dati relativi agli spessori e determinazione di LTi 74

Figura 3.16 – Integrazione numerica e stima della probabilità 74

Figura 3.17 – Definizione delle zone di intersezione mediante ECDF 75

Figura 3.18 – Definizione della soglia ed estrazione dei voxel di intersezione supponendo di porre $\alpha = 0.7$ 76

Figura 3.19 – Zone di intersezione individuate per $\alpha = 0.1$ 77

Figura 3.20 – Zone di intersezione individuate per $\alpha = 0.9$ 77

Figura 3.21 – Zone di intersezione individuate per $\alpha = 0.7$ 77

Figura 3.22 – Tre delle sei maschere impiegate nell'algoritmo hit and miss per l'identificazione dei voxel esterni. 80

Figura 3.23 – Triangoli di mesh generati sulla faccia "1" 80

Figura 3.24 – Triangoli di mesh generati sulla faccia "1" e sulla faccia "3" 81

Figura 3.25 – Applicazione dell'operazione di smoothing ad una mesh generica 82

Figura 3.26 – Applicazione dell'operazione di smoothing 82

Figura 3.27 – Dimensioni caratteristiche del triangolo. 83

Figura 3.28 – Fattore di forma dei triangoli della mesh elaborata; il colore blu identifica triangoli con fattore di forma superiore a 0,5. 83

Figura 3.29 – Esempio di due cluster che condividono parte della loro sfera di taglio 84

Figura 3.30 – Esempio di cluster (evidenziato in blu) la cui rimozione da luogo a tre sotto-immagini relative ad altrettanti entità geometriche 3D (evidenziate in rosso) 85

Figura 3.31 – Modello bidimensionale della struttura 3D cluster/entità raster 86

Figura 3.32 – Risultato del processo di dilatazione unitaria del cluster "X" 87

Figura 3.33 – Risultato dell'operazione (I-C) dell'eq. 3.3 87

Figura 3.34 – Immagine finale con entità separate corredate dal dato sul cluster di appartenenza 88

Figura 3.35 – Polilinea iniziale per l'ordinamento dei voxel 90

Figura 3.36 – Curva interpolante sui nodi della polilinea iniziale 90

Figura 3.37 – Curva finale approssimante 91

Figura 3.38 – Origine della polilinea iniziale 92

Figura 3.39 – Origine dei primi due tratti della polilinea iniziale 93

Figura 3.40 – Importanza del EMST 93

Figura 3.41 – Errore dovuto all'assenza di controllo tramite EMST 94

Figura 3.42 – Errata costruzione di una polilinea 95

Figura 3.43 – Corretta costruzione della polilinea 95

Figura 3.44 – Soluzione all'errore dovuto all'assenza di controllo tramite EMST 96

Figura 3.45 – Corretta costruzione della polilinea 96

Figura 3.46 – Posizionamento dei vertici finali 98

Figura 3.47 – Valutazione dell'errore di approssimazione mediante metodo misto 101

Figura 3.48 – Effetto dell'applicazione del metodo delle funi 102

Figura 3.49 – Generazione delle curve finali 104

Figura 3.50 – Estrazione del set di cicli 106

Figura 3.51 – Estrazione del set di cicli 107

Figura 3.52 – Estrazione del set di cicli 107

Figura 3.53 – Esempio di proiezioni ortogonali la cui traduzione nello spazio 3D non è univoca 108

Figura 3.54 – Proiezioni bidimensionali del modello di Figura 3.51 109

Figura 3.55 – Proiezioni bidimensionali del modello di Figura 3.52 110

Figura 3.56 – Rappresentazione matriciale della lista dei cicli 111

Figura 3.57 – Primo esempio di determinazione di una base di cicli e sua notazione matriciale 112

Figura 3.58 – Secondo esempio di determinazione di una base di cicli e sua notazione matriciale 112

Figura 3.59 – Varie fasi di determinazione del set di cicli finale; l'ultima fase deve essere completata manualmente poiché la base ottimale non prevede per definizione la "chiusura" del modello. 114

Figura 3.60 – Patch finali 114

Figura 3.61 – Dettaglio delle patch interne 115

Figura 3.62 – Patch errata (evidenziata in rosso) 115

Figura 3.63 – Modello wireframe semplificato 116

Figura 4.1 – Dato iniziale sotto forma di immagine raster bidimensionale 117

Figura 4.2 – Estrazione del modello wireframe 3D raster (in verde) e dei vertici 3D 118

Figura 4.3 – Fitting finale delle curve spline tridimensionali 118

Figura 4.4 – Modello wireframe vettoriale 3D – vista "A" 119

Figura 4.5 – Modello wireframe vettoriale 3D – vista "B" 119

Figura 4.6 – Processo di identificazione della base di cicli ottimale 120

Figura 4.7 – Costruzione delle patch di superfice in ambiente Rhinoceros® 120

Figura 4.8 – Risultato finale della procedura di riscostruzione 121

Figura 4.9 – Caso studio "A" – proiezioni ortogonali e bounding box 122

Figura 4.10 – Caso studio "A" – Immagine 3D e Modello CAD 3D 123

Figura 4.11 – Caso studio "B" – proiezioni ortogonali 124

Figura 4.12 – Caso studio “B” – Wireframe 3D 125

Figura 4.13 – Caso studio “B” – Modello CAD 3D 125

Figura 4.14 – Caso studio “C” – proiezioni ortogonali 126

Figura 4.15 – Caso studio “C” – Wireframe 3D 126

Figura 4.16 – Caso studio “C” – Modello CAD 3D 127

Figura 4.17 – Caso studio “D” – Bozzetti 2D 127

Figura 4.18 – Caso studio “D” – Wireframe 3D 128

Figura 4.19 – Caso studio “D” – Modello CAD 3D 128

Figura 4.20 – Caso studio “E” – Bozzetti 2D 129

Figura 4.21 – Caso studio “E” – Wireframe 3D 130

Figura 4.22 – Caso studio “E” – Modello CAD 3D 130

Figura 4.23 – Elaborazione dell’immagine acquisita in ambiente GIMP 132

Figura 4.24 – Posizionamento delle immagini sui piani coordinati in ambiente Rhinoceros® 132

Figura 4.25 – Generazione del modello wireframe 3D in ambiente Rhinoceros® . 133

Figura 4.26 – Generazione del modello 3D a superfici in ambiente Rhinoceros® .. 133

Bibliografia

[1] M.D.G. Ellen Yi-luen Do, Drawing as a means to design reasoning, in: Visual Representation, Reasoning and Interaction in Design Workshop Notes, Artificial Intelligence in Design'96 (AID'96), Stanford University, 1996.

[2] K.P.H. Robert C. Zeleznik, SKETCH: An Interface for Sketching 3D Scenes, in: Rushmeier, Holly (Ed.), SIGGRAPH'96, Addison Wesley, 1996: pp. 163–170.

[3] A. Henzen, N. Ailenei, F. Di Fiore, F. Van Reeth, J. Patterson, Sketching with a low-latency electronic ink drawing tablet, in: Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia - GRAPHITE '05, ACM Press, New York, New York, USA, 2005: p. 51.

[4] P. Michalik, D.H. Kim, B.D. Bruderlin, Sketch- and constraint-based design of B-spline surfaces, in: Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications - SMA '02, ACM Press, New York, New York, USA, 2002: p. 297.

[5] F. Durand, An invitation to discuss computer depiction, in: Proceedings of the Second International Symposium on Non-photorealistic Animation and Rendering - NPAR '02, ACM Press, New York, New York, USA, 2002: p. 111.

[6] L.B. Kara, K. Shimada, Supporting Early Styling Design of Automobiles Using Sketch-based 3D Shape Construction, Computer-Aided Design and Applications. 5 (2008) 867–876.

[7] C. Tian, M. Masry, H. Lipson, Physical sketching: Reconstruction and analysis of 3D objects from freehand sketches, Computer-Aided Design. 41 (2009) 147–158.

[8] M. Idesawa, A System to Generate a Solid Figure from Three View, Bulletin of JSME. 16 (1973) 216–225.

[9] G. Lafue, Recognition of Three-Dimensional Objects from Orthographic Views, Proceedings 3rd Annual Conference on Computer Graphics, Interactive Techniques, and Image Processing, ACWSIGGRAPH, July. (1976) 103–108.

[10] M.A. Wesley, G. Markowsky, Fleshing out wire frames, IBM Journal of Research and Development. 24 (1980) 582–597.
[11] M. Wesley, G. Markowsky, Fleshing out projections, IBM Journal of Research and Development. 25 (1981) 934–954.
[12] R. Lequette, Automatic construction of curvilinear solids from wireframe views, Computer-Aided Design. 20 (1988) 171–180.
[13] I.V. Nagendra, U.G. Gujar, 3-D objects from 2-D orthographic views—A survey, Computers & Graphics. 12 (1988) 111–114.
[14] U.G. Gujar, I.V. Nagendra, Construction of 3D solid objects from orthographic views, Computers & Graphics. 13 (1989) 505–521.
[15] Q.-W. Yan, C.L.P. Chen, Z. Tang, Efficient algorithm for the reconstruction of 3D objects from orthographic projections, Computer-Aided Design. 26 (1994) 699–717.
[16] B.-S. Shin, Y.G. Shin, Fast 3D solid model reconstruction from orthographic views, Computer-Aided Design. 30 (1998) 63–76.
[17] M.H. Kuo, Reconstruction of quadric surface solids from three-view engineering drawings, Computer-Aided Design. 30 (1998) 517–527.
[18] H. Sakurai, D.C. Gossard, Solid model input through orthographic views, ACM SIGGRAPH Computer Graphics. 17 (1983).
[19] S. Liu, Reconstruction of curved solids from engineering drawings, Computer-Aided Design. 33 (2001) 1059–1072.
[20] L. Shixia, H. Shimin, S. Jiaguang, Two accelerating techniques for 3D reconstruction, Journal of Computer Science and Technology. 17 (2002).
[21] C.-F. You, S.-S. Yang, Reconstruction of curvilinear manifold objects from orthographic views, Computers & Graphics. 20 (1996) 275–293.
[22] K. Preiss, Constructing the solid representation from engineering projections, Computers & Graphics. 8 (1984) 381–389.
[23] B.-S. Oh, C.-H. Kim, Systematic reconstruction of 3D curvilinear objects from two-view drawings, Computers & Graphics. 23 (1999) 343–352.
[24] J.A. Brewer, S.M. Courter, Automated conversion of curvilinear wire-frame models to surface boundary models; a topological approach, ACM SIGGRAPH Computer Graphics. 20 (1986) 171–178.
[25] K. Gu, Z. Tang, J. Sun, Reconstruction of 3D objects form orthographic projections, Computer Graphics Forum. 5 (1986).
[26] M. Tanaka, K. Iwama, A. Hosoda, T. Watanabe, Decomposition of a 2D assembly drawing into 3D part drawings, Computer-Aided Design. 30 (1998) 37–46.
[27] W. Geng, J. Wang, Y. Zhang, Embedding visual cognition in 3D reconstruction from multi-view engineering drawings, Computer-Aided Design. 34 (2002) 321–336.

[28] J. Gong, G. Zhang, H. Zhang, J. Sun, Reconstruction of 3D curvilinear wire-frame from three orthographic views, Computers & Graphics. 30 (2006) 213–224.
[29] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction To Algorithms, MIT Press, 2001.
[30] L. Piegl, W. Tiller, The NURBS book (2nd ed.), (1997).
[31] L.B. Kara, C.M.D. Eramo, Pen-based Styling Design of 3D Geometry Using Concept Sketches and Template Models, 1 (2006) 6–8.
[32] D.A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Pearson, 2011.
[33] Y. Abdel-Aziz, H. Karara, {Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry}, 1 (1971).
[34] W. Burger, M.J. Burge, Principles of Digital Image Processing: Fundamental Techniques (Google eBook), Springer, 2009.
[35] E.F. Krause, Taxicab Geometry: An Adventure in Non-Euclidean Geometry, Courier Dover Publications, 1987.
[36] T.Y. Kong, A. Rosenfeld, Topological Algorithms for Digital Image Processing (Google eBook), Elsevier, 1996.
[37] C. Lohou, G. Bertrand, Two symmetrical thinning algorithms for 3D binary images, based on PP-simple points, Pattern Recognition. 40 (2007) 2301–2314.
[38] E.W. Cheney, D.D.R. Kincaid, Numerical Mathematics and Computing, Cengage Learning, 2007.
[39] E L Kaplan,Paul Meier, Nonparametric estimation from incomplete observations, Journal of American Statistical Association. 53 (n.d.) 457 – 481.
[40] M. Couprie, G. Bertrand, New characterizations of simple points in 2D, 3D, and 4D discrete spaces., IEEE Transactions on Pattern Analysis and Machine Intelligence. 31 (2009) 637–48.
[41] C. Lohou, G. Bertrand, A 3D 6-subiteration curve thinning algorithm based on -simple points, Discrete Applied Mathematics. 151 (2005) 198–228.
[42] G. Bertrand, M. Couprie, On Parallel Thinning Algorithms: Minimal Non-simple Sets, P-simple Points and Critical Kernels, Journal of Mathematical Imaging and Vision. 35 (2009) 23–35.
[43] G. Bertrand, On P-simple points, Comptes Rendus De l'Académie Des Sciences. Série 1, Mathématique. 321 (1995) 1077–1084.
[44] G. Bertrand, R. Malgouyres, Some topological properties of surfaces in Z3, Journal of Mathematical Imaging and Vision. 11 (1999) 207–221.
[45] K. Palagyi, A 3D fully parallel surface-thinning algorithm, Theoretical Computer Science. 406 (2008) 119–135.

[46] K. Palágyi, A 3D 6-subiteration thinning algorithm for extracting medial lines, Pattern Recognition Letters. 19 (1998) 613–627.
[47] J. Toriwaki, H. Yoshida, Fundamentals of Three-Dimensional Digital Image Processing (Google eBook), Springer, 2009.
[48] M. Meyer, P. Schr, A.H. Barr, Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow, in: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, 1999: pp. 317–324.
[49] I.T. Jolliffe, Principal Component Analysis (Google eBook), Springer, 2002.
[50] R.C. Veltkamp, Closed Object Boundaries from Scattered Points (Google eBook), Springer, 1994.
[51] R.H. Byrd, J.C. Gilbert, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, Mathematical Programming. 89 (2000) 149–185.
[52] R. H.Byrd, M. E.Hribar, JorgeNocedal, An Interior Point Algorithm for Large-Scale Nonlinear Programming, (2006).
[53] R.A. Waltz, J.L. Morales, J. Nocedal, D. Orban, An interior algorithm for nonlinear optimization that combines line search and trust region steps, Mathematical Programming. 107 (2005) 391–408.
[54] S. Bagali, W.N. Waggenspack, A shortest path approach to wireframe to solid model conversion, in: Proceedings of the Third ACM Symposium on Solid Modeling and Applications - SMA ’95, ACM Press, New York, New York, USA, 1995: pp. 339–350.
[55] R. Diestel, Graph Theory (Google eBook), Springer, 2005.
[56] S. Gabelli, Teoria Delle Equazioni E Teoria Di Galois (Google eBook), Springer, 2008.
[57] F. Abbasinejad, P. Joshi, N. Amenta, Surface Patches from Unorganized Space Curves, Computer Graphics Forum. 30 (2011) 1379–1387.
[58] R. Furferi, L. Governi, M. Palai, Y. Volpe, From 2D Orthographic views to 3D Pseudo-Wireframe: An Automatic Procedure, International Journal of Computer Applications IJCA. 5 (2010) 12–17.
[59] R. Furferi, L. Governi, M. Palai, Y. Volpe, 3D model retrieval from mechanical drawings analysis, International Journal of Mechanics. 2 (2011) 91–98.
[60] R. Furferi, L. Governi, M. Palai, Y. Volpe, Multiple Incident Splines (MISs) algorithm for topological reconstruction of 2D unordered point clouds, International Journal of Mathematics and Computers in Simulation. 5 (2011) 171–179.
[61] R. Furferi, L. Governi, M. Palai, Y. Volpe, The reconstruction problem: Integrating different approaches into a systematic procedure for pseudo wireframe retrieval, Journal for Geometry and Graphics. 1 (2011) 79–91.

[62] L. Governi, R. Furferi, M. Palai, Y. Volpe, 3D geometry reconstruction from orthographic views: A method based on 3D image processing and data fitting, Computers in Industry. (2013).

[63] R. Furferi, L. Governi, M. Palai, Y. Volpe, Detecting intersection zones in thread-like 3D voxel clouds: a novel algorithm based on neighbourhood tracing, JOURNAL OF INFORMATION AND COMPUTATIONAL SCIENCE. 10 (2013) 1–8.

[64] R. Furferi, L. Governi, M. Palai, Y. Volpe, From Unordered Point Cloud to Weighted B-Spline - A Novel Pca-Based Method, in: International Conference on Computer Engineering and Applications, 2011: pp. 146–151.

[65] M. Carfagni, R. Furferi, L. Governi, M. Palai, Y. Volpe, “3D Reconstruction Problem”: An Automated Procedure, in: International Conference on Computer Engineering and Applications, 2011: pp. 99–104.

[66] B. Horn, M.J. Brooks, Shape from Shading, MIT Press, 1979.