

Ao8
873

Convenzione tra l'Università degli Studi "Gabriele D'Annunzio" di Chieti-Pescara e il Comune di Pescara, per la realizzazione di un software gestionale di una banca_dati Cartografica (con determina n. 13 del 14-12-2012).

Pierpaolo Palka
Emanuele Cavallini

Storia di Fotorad dal 2009 al 2013

Prefazione di
Carmine D'ottavio



Copyright © MMXVI
ARACNE editrice int.le S.r.l.

www.aracneeditrice.it
info@aracneeditrice.it

via Quarto Negroni, 15
00040 Ariccia (RM)
(06) 93781065

ISBN 978-88-548-9266-8

*I diritti di traduzione, di memorizzazione elettronica,
di riproduzione e di adattamento anche parziale,
con qualsiasi mezzo, sono riservati per tutti i Paesi.*

*Non sono assolutamente consentite le fotocopie
senza il permesso scritto dell'Editore.*

I edizione: aprile 2016

Indice

7 *Prefazione* di Carmine D' Ottavio

9 *Introduzione*

11 **Capitolo I**
Costituzione del software

1.1. Fotorad: piattaforma grafica proprietaria storia dal 2009 al 2014 – 1.2. Realizzazione delle classi – 1.3. Uso delle classi

23 **Capitolo II**
Visualizzazione grafica in 3D: dalla matrice di rotazione alla matrice prospettica

2.1. Premessa – 2.2. Rappresentare oggetti tridimensionali in 2D e 3D – 2.3. Proiezione Parallela – 2.4. Proiezione Prospettica – 2.5. Visualizzazione Grafica nelle tre dimensioni – 2.6. Significato Geometrico della matrice di rotazione – 2.7. Rotazione antioraria attorno all'Asse X (ω) – 2.8. Rotazione oraria attorno all'Asse Y (φ) – 2.9. Rotazione antioraria attorno all'Asse Y (φ) – 2.10. Rotazione antioraria attorno all'Asse Z (k) – 2.11. Moltiplicazione della Matrice di Rotazione Antioraria intorno all'Asse X per la Matrice di Rotazione Oraria intorno all'Asse Y – 2.12. Moltiplicazione della Matrice di Rotazione Antioraria intorno agli Assi XY per la Matrice di Rotazione Antioraria intorno all'Asse Z – 2.13. Matrice Prospettica – 2.14. Moltiplicazione della Matrice di Rotazione Antioraria R_z per la Matrice di Rotazione R_x – 2.15. Moltiplicazione delle Matrici di Rotazione R_z ed R_x per la Matrice delle Simmetrie M_{yz} – 2.16. Moltiplicazione della Matrice di Traslazione T per le Matrici di Rotazione R_z ed R_x e la Matrice delle Simmetrie M_{yz}

49 Capitolo III
Cattura delle entità in 3D

3.1. Condizioni di appartenenza 3D e utilità dell'UCS – 3.2. Gli strumenti di cattura – 3.3. Piano passante per un punto ortogonale o parallelo a una retta nello spazio

57 Capitolo IV
La rappresentazione delle curve nello spazio

4.1. La serie di Taylor – 4.2. Polinomi di Lagrange – 4.3. Il metodo di Lagrange in forma esplicita – 4.4. Polinomi di Hermite – 4.5. Concatenazione di polinomi di terzo grado

89 Capitolo V
Le Superfici

5.1. Riempimento di una superficie – 5.2. Bucatura di una superficie

95 Capitolo VI
I Solidi

6.1. Solido con superfici – 6.2. Solido propriamente detto o con estrusione ed Algoritmi di ombreggiatura

105 Capitolo VII
Evoluzione del Programma Fotorad

7.1. Algoritmo di Weiler – Atherton – 7.2. Gli Shape File – 7.3. Introduzione dell'Algoritmo per il controllo dell'UCS (Utent Coordinate System)

Prefazione
CARMINE D'OTTAVIO*

Ripercorrere la storia di una invenzione o di uno studio fatto, e tradotto in un oggetto concreto da sottoporre all'attenzione degli altri, ritengo sia interessante.

Secondo il mio parere, infatti, è difficile oltre che interessante conoscere le motivazioni che hanno sollecitato l'opera, le strade che hanno portato alla sua realizzazione, gli ostacoli incontrati ed infine lo scopo a cui si voleva tendere.

A questo punto è più efficace, onde apprezzare tutto il lavoro, comparare i risultati ottenuti con un'opera d'arte. Di fronte ad una scultura, o meglio ancora, ad un quadro rimango estasiato per la sua bellezza, apprezzando il dettaglio del suo colore, della pennellata, dei giochi di luce.

Ma per capire completamente l'opera devo studiare il modo in cui sono applicate le regole della prospettiva per un uso razionale dello spazio, come il punto di vista, i punti di fuga, i rapporti proporzionali, la distanza degli oggetti dal piano di quadro e come tali regole siano state tradotte in pennellate eseguite in modo scientifico.

A questo punto mi pongo il problema: perché le due fasi precedentemente introdotte non vengono quasi mai applicate quando ci poniamo di fronte ad un software, che, tutto sommato, possiamo considerare un'opera geniale?

Spesso, ci si ferma solo al suo uso ed all'apprezzamento della gradevolezza della sua interfaccia nonché la facilità d'uso, tralasciando l'aspetto scientifico derivante dalla traduzione dei concetti matematici e geometrici in termini di algoritmi spesso complessi di cui conosciamo solo l'esistenza, ma non il funzionamento.

Questo lavoro ha tenuto impegnati gli autori per tanto tempo, ed ha richiesto una lunga gestazione per comprendere e, successiva

* Dipartimento di Architettura dell'Università degli Studi "Gabriele d'Annunzio" di Chieti-Pescara.

mente, realizzare, innanzi tutto gli algoritmi necessari, scritti in linguaggio Pascal, per far sì che il nostro software fosse da un lato indipendente da librerie di terze parti, facilmente reperibili in commercio, ed in secondo luogo che fosse completo delle funzioni indispensabili per eseguire un disegno automatico in due e tre dimensioni, con l'arricchimento indispensabile dello studio delle luci, per eseguire anche l'illuminazione degli oggetti e delle scene.

Alla fine, sembra che lo scopo sia stato raggiunto, naturalmente, senza "snobbare" quanto la pubblicistica in materia ha già prodotto e immesso nel mercato. Il lavoro è indirizzato prevalentemente ai giovani studenti che, oggi e ancora di più nel futuro, hanno un rapporto stretto con le tecnologie computerizzate.

Quasi sempre nelle nostre scuole, compresa l'università, molto tempo viene speso nell'acquisizione di abilità operative con l'uso di programmi già disponibili sul mercato specialistico. Alla positività di quanto insegnato si contrappone, però, la difficoltà di avere la disponibilità di programmi idonei a risolvere problemi che scaturiscono dalla risoluzione di proprie esigenze, inerenti un lavoro di architettura o di altro genere, come ad esempio il rilievo di facciate prospicienti strade anguste dove la stessa fotogrammetria o lo scanner 3D trovano difficoltà di esecuzione. Già esaminati nel primo libro dal titolo *Storia di Fotorad dal 1984 al 2008*.

Da questa "ratio" è nato il desiderio di individuare una via tecnica per programmare in modo autonomo senza dipendere da altri.

A tale aspetto, da non sottovalutare, si unisce un altro motivo, anch'esso importante: l'aspetto economico. Senza dubbio il mercato è ricco di prodotti informatici, che assemblati insieme, possono addivenire al soddisfacimento degli interessi dell'utenza, ma essi hanno un costo molto elevato. Ecco che l'altro nostro scopo è quello di ridurre le spese per una tecnologia di mercato a vantaggio di un programma di proprietà propria.

A conforto di quanto detto c'è la seconda convenzione con il Comune di Pescara stipulata al fine della realizzazione di un software per la trasposizione del piano regolatore su base catastale, evitando una spesa eccessiva con l'acquisto di un software in commercio che, in ogni caso, non avrebbe risolto un suo problema specifico.

Introduzione

In questo libro viene esposto per la prima volta lo studio fatto per rendere modulare la piattaforma grafica RilArch brevetto UdA 1996 attraverso l'uso delle classi scritte in linguaggio Pascal. In tal modo da programma di disegno automatico a sé stante, RilArch diventa anche la piattaforma grafica comune ai programmi Fotorad, Fotorad GIS in Web e Prg FZone.

Vengono inoltre esposti i principali passaggi per la realizzazione degli algoritmi per il disegno di curve nello spazio, nonché vengono studiati e realizzati gli algoritmi per la cattura degli elementi grafici di 3D.

Vengono ancora esposti il funzionamento degli algoritmi preposti alla rotazione degli elementi geometrici nonché alla loro visualizzazione prospettica spaziale.

Lo studio e la realizzazione degli algoritmi preposti alla illuminazione degli oggetti 3D completa il quadro illustrativo di tutta la tematica inerente la visualizzazione globale degli oggetti spaziali.

Viene infine affrontato il problema del disegno di entità poste su piani differenti da quello tradizionale X Y tramite l'introduzione dell'UCS.

Uno sguardo particolare è stato, poi, riservato all'algoritmo di Clipping tra poligoni concavi. Algoritmo che è stato realizzato proprio per la risoluzione di un problema specifico richiesto dal Comune di Pescara con cui è stata stipulata anche una convenzione, riportata a nota.

Capitolo I Costituzione del software

1.1. Fotorad: piattaforma grafica proprietaria storia dal 2009 al 2014

Nella prima parte della *Storia di Fotorad dal 1984 al 2008* (Aracne Editrice), abbiamo concluso dicendo che era opportuno dividere la storia di Fotorad in due momenti salienti: il primo in cui viene illustrato in ogni minimo particolare il percorso che ne ha caratterizzato la nascita, la crescita e lo sviluppo della nostra ricerca. Il secondo che merita una trattazione a parte e che risulta caratterizzato da situazioni che coinvolgono il modo di programmare con la nota programmazione ad oggetti, attraverso l'uso delle classi oltre che trattare l'aspetto commerciale attraverso le convenzioni con gli Enti Pubblici.

La scommessa di investire sulla realizzazione di un software, nonostante la nostra preparazione in campo informatico non sia votata alla realizzazione di un applicativo nel settore commerciale, ha prodotto delle convenzioni con enti pubblici per il controllo dei dati territoriali. Ricordiamo, infatti, che siamo abituati più a realizzare algoritmi parziali piuttosto che software completi, almeno nel campo della grafica computerizzata, vista le complesse problematiche geometriche, analitiche nonché informatiche, che si celano all'interno di un programma di disegno automatico. Questa collaborazione ci ha fatto comprendere meglio la distinzione tra la realizzazione di una banca dati e la realizzazione di un software gestionale della stessa banca dati: i due aspetti, teorico-pratici, hanno portato a migliorare lo studio teorico i cui risultati, nell'applicazione pratica, evidenziano alcune inadeguatezze. L'elaborazione di una banca dati è basata su normative che ne regolano l'uso e spesso accade nella pratica che ci si trovi davanti a difficoltà di realizzazione dell'idea stessa implicita nel database. Ad esempio, non ci si pone il problema se e in che modo si possa visualizzare un file vettoriale da parte di un utente in rete, difficoltà di visualizzazione dei dati.

Continuando la storia del software Fotorad, va sottolineato che a cavallo degli anni 2008 e 2009 il software trova una seconda applicazione finalizzata alla gestione di dati territoriali, oltre a quella del rilievo architettonico, usato dai professionisti nel campo del restauro e della ristrutturazione di edifici o manufatti in genere. Infatti l'algoritmo di trasformazione omografica usato in Fotorad per il raddrizzamento computerizzato dell'immagine, e già noto a chi ne ha sperimentato la funzionalità, trova una seconda applicazione interessante nella georeferenziazione a quattro punti di carte topografiche, più volte verificato in campo, e che si consolida ampiamente con la stipula, in data 29/11/07, della convenzione tra il Comune di Borrello (CH) e il Dipartimento PRICOS dell'Università degli Studi "Gabriele D'Annunzio" Chieti – Pescara in collaborazione con WWF e Legambiente.

Tale convenzione, oltre ai dettami legati ai contenuti amministrativi e legislativi, è stata espressione di richieste tecnico-operative specifiche e particolari, che hanno impegnato l'autore del software ed i suoi collaboratori ad iniziare un ulteriore profondo cambiamento dello stesso software per renderlo definitivamente modulare alle nuove esigenze. In definitiva occorreva raccogliere a fattore comune tutti gli algoritmi preposti ad una determinata azione.

In effetti fino al momento della convenzione di cui sopra, il software era stato concepito come un solo programma che raccogliesse al suo interno tutte le procedure degli algoritmi inerenti il disegno automatico. All'interno di questo tipo di piattaforma sarebbero stati inseriti necessariamente anche gli algoritmi preposti alla gestione di dati contenuti in appositi Data Base, molte volte con funzioni ripetitive secondo il classico pensiero della programmazione procedurale. La conseguenza più grave era che tutto ciò portava ad una grande confusione nella fase di realizzazione del software preposto alla gestione dei dati territoriali, e, soprattutto, nella sua manutenzione ed aggiornamento nel tempo.

Il software in effetti è composto da due momenti salienti: uno dedicato al disegno automatico e l'altro alla gestione dei dati che in questo andavano a convivere in uno stesso spazio di programmazione. Per progredire nel nostro lavoro occorreva necessariamente cambiare filosofia di programmazione passando alla "programmazione a

classi con istanze di oggetti” con cui era possibile realizzare oggetti con funzioni proprie, pur collaborando sinergicamente con gli altri oggetti. La necessità del cambiamento era indispensabile per non consegnare ad un ente pubblico, come il Comune di Borrello, un software che al suo interno avesse sia funzioni di disegno che funzioni di gestione dei dati, poiché molte volte era necessario intervenire solo sugli algoritmi inerenti la gestione dei dati e pochissime volte sugli algoritmi del disegno automatico.

Ricordiamo, infatti, che il programma di gestione dati territoriali poggia su una piattaforma grafica proprietaria, brevetto UdA 1994/96, che ne regola le funzioni di disegno con la doverosa separazione da esso della parte riguardante la gestione dei dati veri e propri da realizzare con un modulo separato, le cui informazioni potevano essere legate, attraverso una chiave di ricerca, alle geometrie disegnate nella piattaforma grafica.

La suddivisione del software, preposto alla gestione della grafica e a quella dei dati, in un insieme di oggetti singoli sarebbe risultato più semplice ed efficace ed avrebbe consentito di intervenire solo su quello chiamato in causa. Dopo approfonditi studi, pertanto, si è pervenuti allo smembramento del programma Fotorad che è stato suddiviso in classi, indipendenti fra loro, ciascuna delle quali preposta ad un compito specifico.

Il software è stato diviso in due parti fondamentali: la cosiddetta piattaforma grafica che si sarebbe occupata solo del disegno automatico con tutte le funzionalità proprie del disegno. Da questa piattaforma sono scaturiti nel tempo tre software specifici nella progettazione CAD – Rilarch – disegno digitalizzato da immagine per l’architettura – Fotorad – e disegno digitalizzato per il territorio – Fotorad GIS.

1.2. Realizzazione delle classi

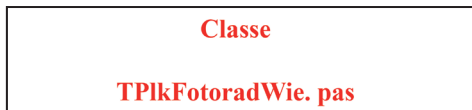
La maggiore conoscenza dei linguaggi di programmazione e la necessità di avviare un progetto mirato al continuo sviluppo di aggiornamento della stessa piattaforma grafica hanno determinato una importante svolta in merito alla rinnovata realizzazione del

software, interamente ristrutturato e ricompilato secondo la tecnica della programmazione ad oggetti, rendendo le classi responsabili di ciascun settore particolare.

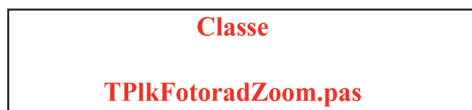
L'aver dato seguito a questa filosofia ha fatto ideare il nuovo progetto Fotorad, articolato secondo il sottostante schema illustrativo, nostro primo programma ad oggetti denominato: Fotorad 2009.



È responsabile della registrazione delle coordinate di tutte le entità grafiche appartenenti sia alla finestra immagine che alla finestra restituzione vettoriale, nonché delle proprietà ad esse associate.



Discende dalla classe *TPlkFotoradCoord* ed è responsabile della sola rigenerazione sia della finestra immagine che di quella vettoriale.



Discende dalla precedente Classe *TPlkFotoradWie.pas* ed è responsabile degli ingrandimenti, delle riduzioni e degli spostamenti sia delle entità appartenenti alla finestra immagine, sia di quelle appartenenti alla finestra di restituzione.



Discende dalla precedente Classe *TPlkFotoradZoom.pas* ed è responsabile della cattura delle entità sia nella finestra immagine che in quella di restituzione.



Discende da Classe *TPlkFotoradCat.pas* ed essa rappresenta la classe che eredita tutte le caratteristiche delle classi precedenti. In essa sono definiti i comandi mirati al disegno (punto, linea, polilinea, etc, etc,) ed al controllo delle azioni applicabili a ciascuna entità (taglia, estendi, etc.).

La classe *TPlkCoord* è quella fondamentale. In essa sono state definite tutte le variabili tipo adibite alla creazione dei dati geometrici e le proprietà di ogni singola entità. Ad esempio:

Punto[0] = Cod0, rosso, 2, Piano_0, etc.
 Linea[0] = Cod1, verde, 1, Piano_0, etc.
 Polilinea[0] = Cod2, giallo, 3, Piano_0 etc.

Punto[1] = Cod0, rosso, 2, Piano_1, etc.
 Linea[1] = Cod1, verde, 1, Piano_1, etc.
 Polilinea[1] = Cod2, giallo, 3, Piano_1 etc.

Punto[n] = Cod0, rosso, 2, Piano_1, etc.
 Linea[n] = Cod1, verde, 1, Piano_1, etc.
 Polilinea[n] = Cod2, giallo, 3, Piano_1 etc.

La stessa classe, però, utilizzata da sola non ha molto senso poiché non contiene nessuna routine adibita alla gestione degli elementi geometrici.

Essa, infatti, va unita alla classe *TPlkFotoradView* che eredita tutte le proprietà della precedente. In essa è definita quella procedura ritenuta basilare per un programma di disegno automatico. Ovvero la procedura adibita alla rigenerazione di tutte le entità geometriche archiviate in un Data-Base sequenziale distinte e nominate con un codice proprio.