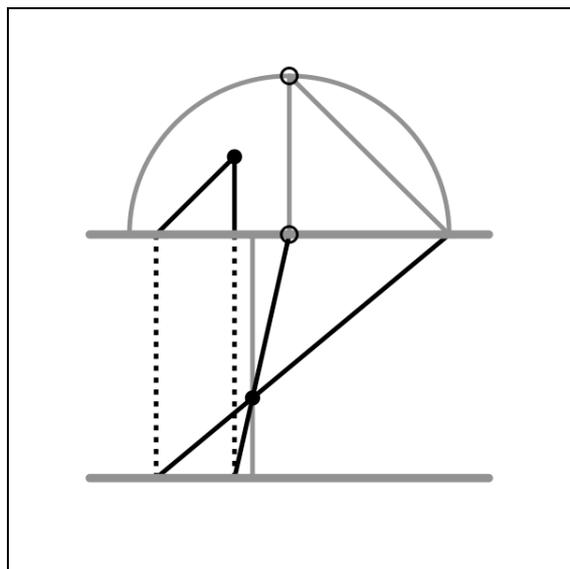


FRANCESCO VINCI

***BRUNELLESCHI***

Software per la restituzione prospettica assistita



Tesi di Dottorato di Ricerca in  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
XIX CICLO

*Università degli Studi di Catania, Facoltà di Architettura*  
**Tutor, Prof. Arch. Giuseppe Pagnano**

SIRACUSA, OTTOBRE 2006

## *Indice*

Introduzione al tema di ricerca	7
<b>FONDAMENTI SCIENTIFICI</b>	
Fotogrammetria e restituzione prospettica	11
La restituzione prospettica con <i>Brunelleschi</i> , potenzialità e limiti	35
Determinazione dell'orientamento della prospettiva	37
Metodo di restituzione	45
Cenni di programmazione	47
<b>BRUNELLESCHI MANUALE OPERATIVO</b>	
Licenza d'uso del software	55
Requisiti di sistema, installazione, informazioni preliminari	57
Importazione della rappresentazione prospettica	61
Determinazione dell'orientamento della prospettiva	63
Definizione degli elementi geometrici della prospettiva	69
Gestione del modello restituito	78
<b>ESEMPI DI RESTITUZIONE</b>	
Restituzione da una rappresentazione prospettica	91
Restituzione da una prospettiva di Andrea Pozzo	97
RESTITUZIONI PROSPETTICHE ASSISTITE	104
RIPRODUZIONE DELLE TAVOLE GRAFICHE	117
STRALCI DEL CODICE DI PROGRAMMAZIONE	139
<b>BIBLIOGRAFIA</b>	

### *Introduzione al tema di ricerca*

*Brunelleschi*, software di restituzione prospettica assistita al computer, rappresenta il risultato finale, e tesi, del mio percorso di ricerca per il dottorato in Teoria e Storia della Rappresentazione, coordinato dal prof. arch. Giuseppe Pagnano alla Facoltà di Architettura di Siracusa, Università degli Studi di Catania, XIX Ciclo.

Il nome che ho scelto, *Brunelleschi*, è evocatore di prospettiva e della scientificità di questo metodo di rappresentazione, verso cui nutro curiosità e interesse fin dai primi studi liceali e universitari, così come nei confronti del procedimento "inverso" di restituzione prospettica.

Per il raggiungimento dell'obiettivo prefissatomi, dopo averne selezionato le potenzialità e i limiti, ho studiato alcuni procedimenti grafici della restituzione prospettica, e ho approfondito le mie conoscenze nel campo della programmazione dei personal computer; detti procedimenti grafici sono stati tradotti in formule note di geometria analitica, successivamente trascritte e codificate secondo la sintassi del linguaggio di programmazione prescelto -il *RealBasic*- e coerentemente inserite nella struttura gerarchica del *software*.

*Brunelleschi* non è stato sviluppato come applicativo "invadente", basato prevalentemente su automatismi gestiti direttamente dal *software* e poco controllabili dall'utente finale, bensì come strumento affidato alla guida sapiente di chi se ne vorrà servire, una sorta "pantografo" della restituzione prospettica; a partire da un'immagine digitale, acquisita tramite scanner o altro dispositivo, sarà compito dell'operatore sintetizzare e individuare, sulla veduta prospettica, dapprima l'orientamento della prospettiva, successivamente gli elementi geometrici -linee, piani, volumi- che il programma si occuperà di "restituire" nella loro forma "reale" attraverso un modello tridimensionale digitale.

Lo sviluppo del *software Brunelleschi* è stato basato sul continuo confronto e sperimentazione su prospettive scientifiche, di cui era nota quanto meno la *pianta*, o proiezione orizzontale, attraverso la verifica della corrispondenza di questa con la proiezione di Monge dal modello tridimensionale “restituito” dal *software*.

Il linguaggio di programmazione utilizzato, *RealBasic*, è del tipo *object-oriented* ed *event-driven*, ha il vantaggio di compilare applicativi multipiattaforma, ed inoltre offre un metodo di sviluppo attraverso il quale ciascuna parte del *software*, che si occupa di una specifica funzione, costituisce un modulo integrato con gli altri ma distinto e separato, così da poter essere facilmente oggetto di revisione o modifica, e favorire una costante ottimizzazione del codice.

Il risultato finale dello studio e del lavoro svolto, *Brunelleschi*, è, o può essere, allo stato attuale, *oggetto* di ulteriori percorsi di sviluppo, miglioramento, accrescimento delle sue stesse potenzialità, così come *strumento* di ricerca e di analisi grafica sulle rappresentazioni prospettiche.

*Brunelleschi* è un *software* gratuito -il suo utilizzo è semplicemente sottoposto alla richiesta di un codice di autorizzazione- liberamente scaricabile dal sito *internet* <http://www.arcobaleno2001.it/> nella sua versione più recente, selezionando sulla sinistra il collegamento relativo (voce *Brunelleschi*).

*FONDAMENTI SCIENTIFICI*





## FOTOGRAMMETRIA E RESTITUZIONE PROSPETTICA

La "Fotogrammetria" è un metodo che consente di desumere le dimensioni di un oggetto attraverso una o più immagini fotografiche, basandosi sui principi della geometria proiettiva e sulla congruenza fra immagine fotografica e immagine prospettica. Tale metodo fu ideato alla fine dell'800 dal colonnello francese Laussedat, e sperimentato a livello topografico e su opere architettoniche.

Nella maggior parte dei casi, il procedimento di restituzione fotogrammetrica si avvale di coppie di immagini fotografiche da diversi punti di vista reciprocamente noti; come è intuibile -considerato che l'immagine prospettica è individuata sul "quadro" dall'intersezione sullo stesso del cono visivo, ottenuto proiettando ciascun punto dell'oggetto osservato dal vertice, centro di vista dell'osservatore- ciascun punto rappresentato in un'immagine prospettica è anche rappresentazione degli infiniti punti che appartengono al medesimo raggio proiettante dal centro di vista; ne discende che per determinarne univocamente la posizione reale, occorre servirsi, in assenza di altri dati, di una seconda immagine fotografica da diverso punto di vista; l'intersezione dei due raggi visuali individua univocamente il punto cercato. Tale tecnica, associata all'ausilio fornito da apparecchiature apposite, è utilizzata principalmente nel rilievo "aerofotogrammetrico".

Avendo a disposizione una sola immagine fotografica o prospettica, è tuttavia possibile, se sono noti alcuni opportuni dati metrici, eseguire la "Restituzione Prospettica", avvalendosi di procedimenti noti di geometria descrittiva, inversi rispetto a quelli del disegno in prospettiva. In questo senso intesa, l'automatizzazione della restituzione prospettica è l'ambito di sviluppo prescelto dell'applicativo per personal computer *Brunelleschi*, tema di questa tesi di dottorato, mentre è

noto che applicazioni *software* fotogrammetriche sono da tempo disponibili, la maggior parte delle quali rivolte al cosiddetto “raddrizzamento dei prospetti” e pertanto limitate a questo specifico compito.

Fotogrammetria e restituzione prospettica, quali strumenti di analisi grafica e di rilievo, mantengono la loro importanza e attualità pur se affiancati da nuove e moderne tecniche di rilievo diretto, quali ad esempio il *laser-scanner*; è persino superfluo affermare che il rilievo diretto può riguardare ciò che è esistente, mentre riguardo ciò che non esiste, fra le opere architettoniche, perché andato distrutto o rimasto sulla carta, una sola immagine prospettica può essere base sufficiente per procedere alla restituzione della realtà metrica.

Anna Sgrosso descrive i concetti basilari della fotogrammetria e illustra i procedimenti geometrici della restituzione prospettica; «Nello studio dei procedimenti necessari per eseguire una restituzione prospettica, appare prioritaria la conoscenza degli elementi essenziali, della nomenclatura generalmente adottata e delle differenze tra una prospettiva geometrica e la corrispondente immagine fotografica. Tutte le operazioni che saranno indicate sono volte ad individuare la posizione del centro di vista (obiettivo della macchina fotografica) rispetto al piano dell'immagine (pellicola o lastra) e all'oggetto rappresentato: in sostanza a corredare il fotogramma degli elementi di riferimento.

Si è già accennato in principio alle relazioni proiettive intercorrenti fra gli enti dello spazio (punti, rette, piani) e le relative immagini: tali enti nella rappresentazione prospettica vengono riferiti generalmente ad un piano ausiliario (quello di calpestio) che si aggiunge ai tre elementi fondamentali del riferimento: il *quadro*  $\pi$ , che coincide con il piano dell'immagine; il *centro di vista*  $V$ ; la *distanza principale*  $d$ , che è quella del centro di vista dal quadro e corrisponde alla distanza focale dell'apparecchio fotografico. Nella prospettiva geometrica il piano

ausiliario di riferimento, denominato *geometrico*, viene rappresentato, secondo le norme del metodo delle proiezioni centrali, mediante due rette parallele la *fondamentale f* (intersezione del geometrico con il quadro) e la retta di orizzonte *o* (immagine della retta all'infinito, o retta impropria, del geometrico stesso e di tutti i piani orizzontali); quest'ultima è l'intersezione col quadro del *piano di orizzonte*, condotto per il centro di vista e parallelo al geometrico. Fondamentale e retta di orizzonte costituiscono rispettivamente la traccia e la retta di fuga del piano geometrico e la loro mutua distanza viene assimilata all'altezza dell'osservatore»<sup>1</sup>.

Il procedimento grafico di restituzione di una prospettiva geometrica alla sua realtà metrica, nella scala del riferimento, discende con evidenza dalla considerazione della corrispondenza omologica -sul quadro- fra il ribaltamento della vera forma dal geometrico e deformata prospettica, con la fondamentale quale asse dell'omologia, il ribaltamento del punto di vista sul quadro rispetto all'orizzonte come centro, e l'orizzonte stesso come retta limite.

«Nella prospettiva fotografica il quadro è costituito ... dalla pellicola (o dalla lastra), il centro di vista dal centro dell'obiettivo e la distanza principale dalla distanza focale dell'apparecchio. A differenza della prospettiva geometrica, in cui il quadro si suppone situato tra l'osservatore e l'oggetto da rappresentare, nella prospettiva fotografica oggetto e centro di vista si trovano dalla stessa parte rispetto al piano della pellicola: per questo motivo l'immagine fotografica risulta capovolta e, data la piccola distanza della lastra dall'obiettivo, anche notevolmente rimpicciolita. E' dunque lecito considerare l'immagine fotografica di un oggetto congruente ad una prospettiva dell'oggetto stesso costruita scegliendo il quadro in posizione simmetrica a quella detta pellicola rispetto al centro di vista, quindi servirsi di essa per eseguire le operazioni di restituzione prospettica»<sup>2</sup>.

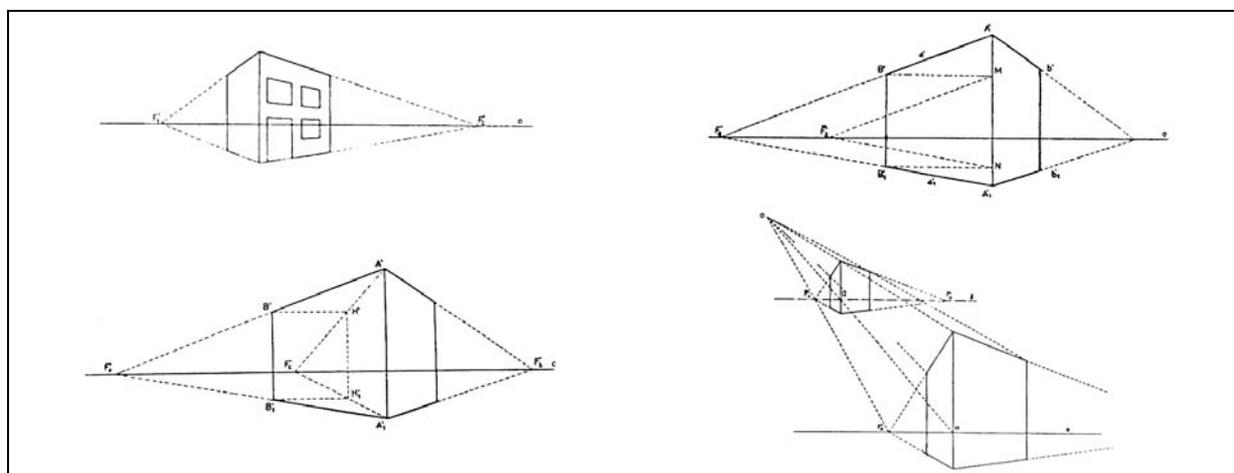
<sup>1</sup> Anna Sgrosso, *Note di Fotogrammetria Applicata all'Architettura*, Napoli, 1979, pp.8-9; dallo stessa op.cit. sono tratte le figure nelle pagg.14-25 di questo testo

<sup>2</sup> Ivi pp.10-11

Per poter procedere alla restituzione, occorre determinare gli elementi di "orientazione interna" (centro di vista e distanza principale) e di "orientazione esterna" (retta fondamentale); sempre Anna Sgrosso ne descrive i procedimenti, dapprima riferendosi alle prospettive a quadro verticale, per poi estendere la trattazione alle prospettive a quadro inclinato. Il primo passo da compiere è la determinazione della retta di orizzonte, la quale «in quanto retta di fuga del piano geometrico e di tutti i piani ad esso paralleli, ... è pure il luogo dei punti di fuga di tutte le rette orizzontali»<sup>3</sup>; si pone l'esempio della rappresentazione prospettica di un edificio: «se la prospettiva è accidentale, ciò che si verifica quando nessuna faccia dell'edificio è parallela al quadro, ... scelte le immagini (non parallele) di due spigoli orizzontali di una stessa faccia dell'edificio, si prolunghino fino al loro punto comune (di) fuga di tali rette: in quanto tale, questo punto appartiene alla costruenda retta di orizzonte, che sarà dunque la perpendicolare agli spigoli orizzontali (sic) condotta per esso»<sup>4</sup>. Essendo il testo rivolto a procedimenti grafici manuali, vengono suggerite modalità per superare i limiti imposti dal foglio di lavoro cartaceo; vengono illustrati il metodo dei punti di fuga "fittizi" o "ridotti", il metodo delle rette ausiliarie e il metodo della riduzione omotetica, qui mostrati in figura sottostante.

<sup>3</sup> *Ivi* pp.12-13

<sup>4</sup> *Ibidem*

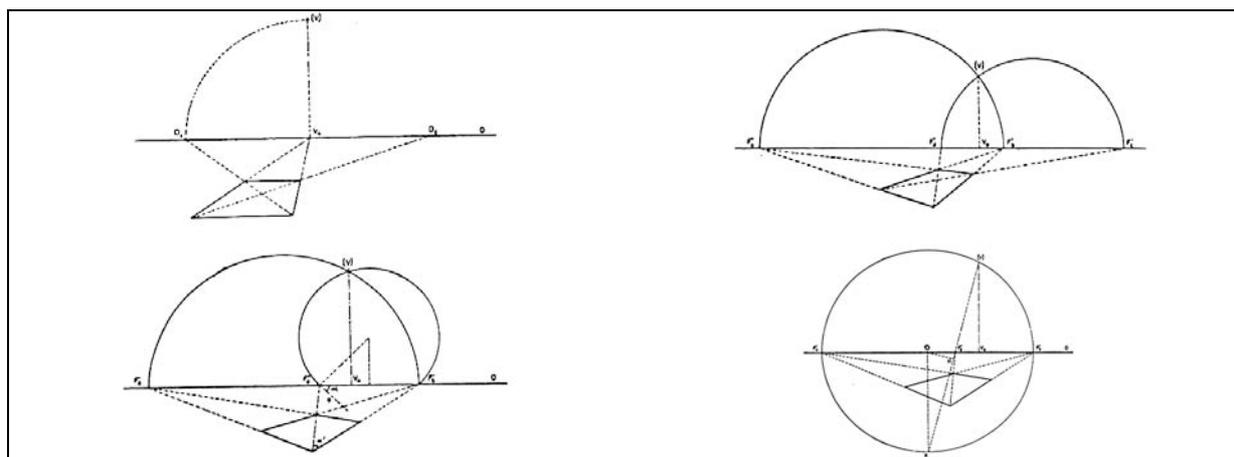


La determinazione dell'orientazione interna, ossia del centro di vista e della distanza principale, è possibile se sono note le dimensioni reali di alcuni elementi rappresentati prospetticamente; i primi metodi grafici presi in considerazione si basano sulla presenza di un quadrato nella rappresentazione prospettica; «se nel fotogramma è compresa l'immagine di un quadrato appartenente al piano geometrico o ad altro piano orizzontale, e in essa due lati opposti sono paralleli, i corrispondenti lati del quadrato obiettivo sono paralleli al quadro; le immagini degli altri due lati, ortogonali ai primi, concorrono allora nel punto ... (di) fuga delle rette perpendicolari al quadro»<sup>5</sup> (“punto principale” sul quadro). «Determinato dunque (tale punto), come intersezione delle rette cui appartengono i lati non paralleli del quadrilatero-immagine, ... la distanza principale si ottiene sulla retta di orizzonte determinando su di essa uno dei *punti di distanza* (punti di fuga delle rette orizzontali inclinate al quadro di 45°): prolungata una delle diagonali del quadrilatero-immagine e individuato ad esempio il punto di distanza sinistro»<sup>6</sup>, la distanza principale è pari al segmento che lo congiunge al punto principale. Come facilmente intuibile, «è questo il caso più semplice che possa presentarsi in una restituzione, in quanto la sola presenza del quadrato già fornisce la conoscenza di due angoli; quello delle due direzioni dei lati (di 90°) e quello di ciascuna diagonale con i lati stessi (di 45°), oppure delle due diagonali (ancora di 90°)»<sup>7</sup>. Se la prospettiva del quadrato è accidentale, il centro di vista e la distanza principale sono determinabili individuando due semicirconferenze, avente la prima come diametro la distanza fra i punti di fuga di due lati ortogonali, e la seconda la distanza fra i punti di fuga delle diagonali; l'intersezione fra le due semicirconferenze è il ribaltamento sul quadro del centro di vista, e la sua distanza dall'orizzonte è pertanto la distanza principale.

<sup>5</sup> *Ivi* pp.18-19

<sup>6</sup> *Ibidem*

<sup>7</sup> *Ibidem*



Se, effettuando la costruzione manualmente, il punto di fuga di una delle diagonali cade fuori del foglio del disegno, sono noti (in figura sovrastante) procedimenti grafici alternativi che consentono di determinare gli elementi di orientazione interna cercati.

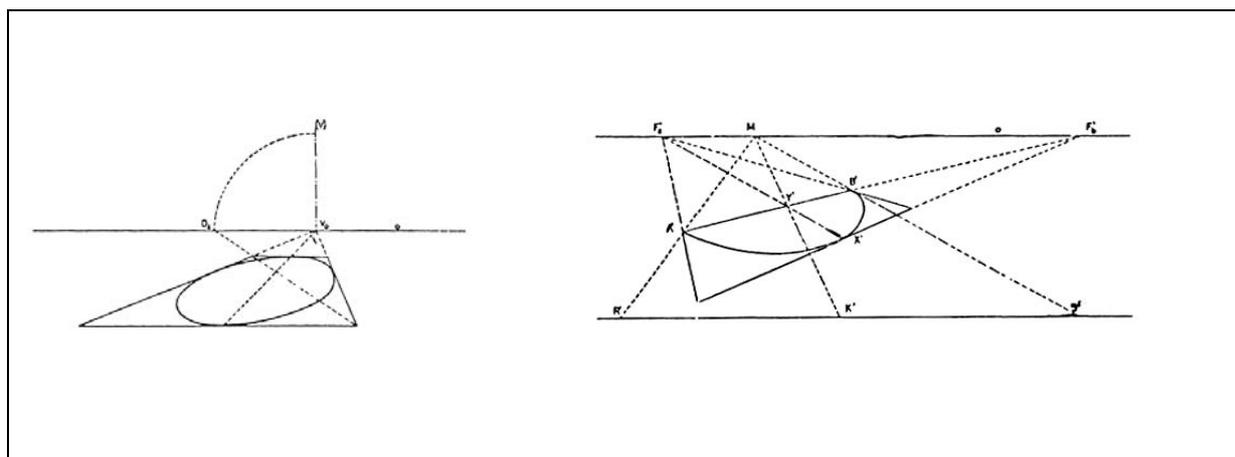
Qualora nella rappresentazione è presente la deformata prospettica di un cerchio appartenente ad un piano orizzontale, «è possibile ... circoscrivere l'ellisse-immagine in un quadrilatero, che sia la rappresentazione prospettica di un quadrato circoscritto al cerchio e che abbia due lati paralleli al quadro e quindi»<sup>8</sup> all'orizzonte, che deve essere pre-determinato. «Si traccino pertanto le due tangenti all'ellisse parallele (alla retta d'orizzonte): la congiungente i due punti di contatto, quale immagine di una retta perpendicolare al quadro, interseca»<sup>9</sup> la retta d'orizzonte nel punto principale sul quadro, «mentre le tangenti condotte all'ellisse»<sup>10</sup> da detto punto principale «forniscono le immagini degli altri due lati del quadrato. Anche qui, tracciata una delle diagonali, resta determinato sulla retta di orizzonte il relativo punto di fuga ... e quindi il segmento ... uguale alla distanza principale»<sup>11</sup>. Se nella rappresentazione è presente una semiellisse, deformata prospettica di un semicerchio, appartenente ad un piano orizzontale, è possibile costruire il rettangolo circoscritto ed utilizzare il procedimento grafico descritto di seguito.

<sup>8</sup> *Ivi* p.23

<sup>9</sup> *Ibidem*

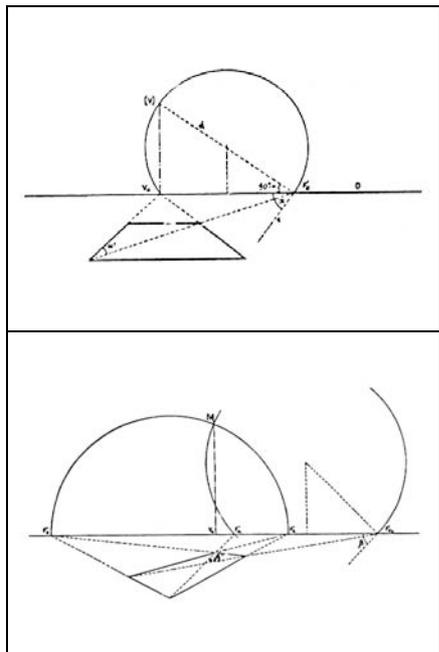
<sup>10</sup> *Ibidem*

<sup>11</sup> *Ibidem*



Dato un rettangolo orizzontale, «è necessario conoscere il rapporto dei lati oggettivi, perché mediante questo è possibile ricavare gli angoli delle diagonali o quelli che esse formano con i lati. Data dunque la prospettiva di un tale rettangolo, se questa ha due lati paralleli, i corrispondenti lati del rettangolo obiettivo sono paralleli al quadro e quindi gli altri due ... hanno le immagini concorrenti»<sup>12</sup> nel punto principale sul quadro, determinando, oltre allo stesso, la retta d'orizzonte. Intercettato su questa il punto di fuga della diagonale, il ribaltato sul quadro del centro di vista (e quindi la distanza principale) si determina quale intersezione fra la perpendicolare alla retta d'orizzonte, condotta dal punto principale, e l'arco di circonferenza avente come corda il segmento che congiunge punto principale e punto di fuga della diagonale, secondo la costruzione mostrata in figura nella pagina successiva, dato l'angolo che la diagonale forma con i lati del rettangolo, essendo noto il rapporto fra questi. Tale costruzione può essere semplificata evitando il tracciamento della circonferenza, cioè intersecando la perpendicolare all'orizzonte, tracciata dal punto principale, con una retta condotta dal punto di fuga della diagonale tale che con l'orizzonte formi un angolo uguale a quello reale che la stessa diagonale forma con i lati del rettangolo paralleli al quadro, come risulta evidente.

<sup>12</sup> Ivi p.25



Nel caso in cui il rettangolo sia in prospettiva accidentale, il ribaltato sul quadro del centro di vista è determinato dall'intersezione dell'arco di circonferenza capace dell'angolo delle due diagonali, costruito sul segmento che congiunge i punti di fuga delle stesse, con la semicirconferenza avente come diametro i punti di fuga dei lati del rettangolo.

«Se il fotogramma non consente l'adozione di alcuna delle costruzioni precedenti ... si può ricorrere a una figura contenuta nel piano verticale»<sup>13</sup>; Anna Sgrosso precisa che «ogni piano verticale, cioè perpendicolare al *geometrico*, si rappresenta in prospettiva mediante una coppia di rette (*fuga e traccia*) perpendicolari alla retta di orizzonte; se il piano è anche ortogonale al quadro (*piano di profilo*) la retta di fuga passa per il punto  $V_0$ »<sup>14</sup> (punto principale). Data la prospettiva di un quadrato contenuto in un piano verticale di profilo, («le immagini di due lati sono parallele, quei due lati sono paralleli al quadro e quindi verticali»<sup>15</sup>), le immagini dei due lati non verticali, in quanto perpendicolari al quadro, concorrono nel punto principale. Il centro di vista «si determina sulla retta di fuga del piano di profilo ... quale punto di fuga di una delle diagonali del quadrato, in questo caso inclinate al quadro di un angolo di  $45^\circ$ : infatti il cerchio di distanza, cui appartiene»<sup>16</sup> il ribaltato sul quadro del centro di vista «è anche il luogo dei punti di fuga di tutte le rette dello spazio inclinate al quadro di  $45^\circ$ »<sup>17</sup>. Nel caso in cui il quadrato appartenga ad un piano verticale non ortogonale al quadro, occorre che sia noto un angolo orizzontale, ad esempio «ci si può servire dell'angolo delle due direzioni ortogonali dell'edificio»<sup>18</sup>. Sulla rappresentazione prospettica, si determina e disegna la retta di fuga del piano verticale che contiene il quadrato, e si interseca con il prolungamento di una delle diagonali (in figura sovrastante); si costruisce l'arco di cerchio avente come corda il segmento che unisce tale intersezione con il punto di fuga dei lati non verticali del

<sup>13</sup> *Ivi* pp.27-28

<sup>14</sup> *Ibidem*

<sup>15</sup> *Ibidem*

<sup>16</sup> *Ibidem*

<sup>17</sup> *Ibidem*

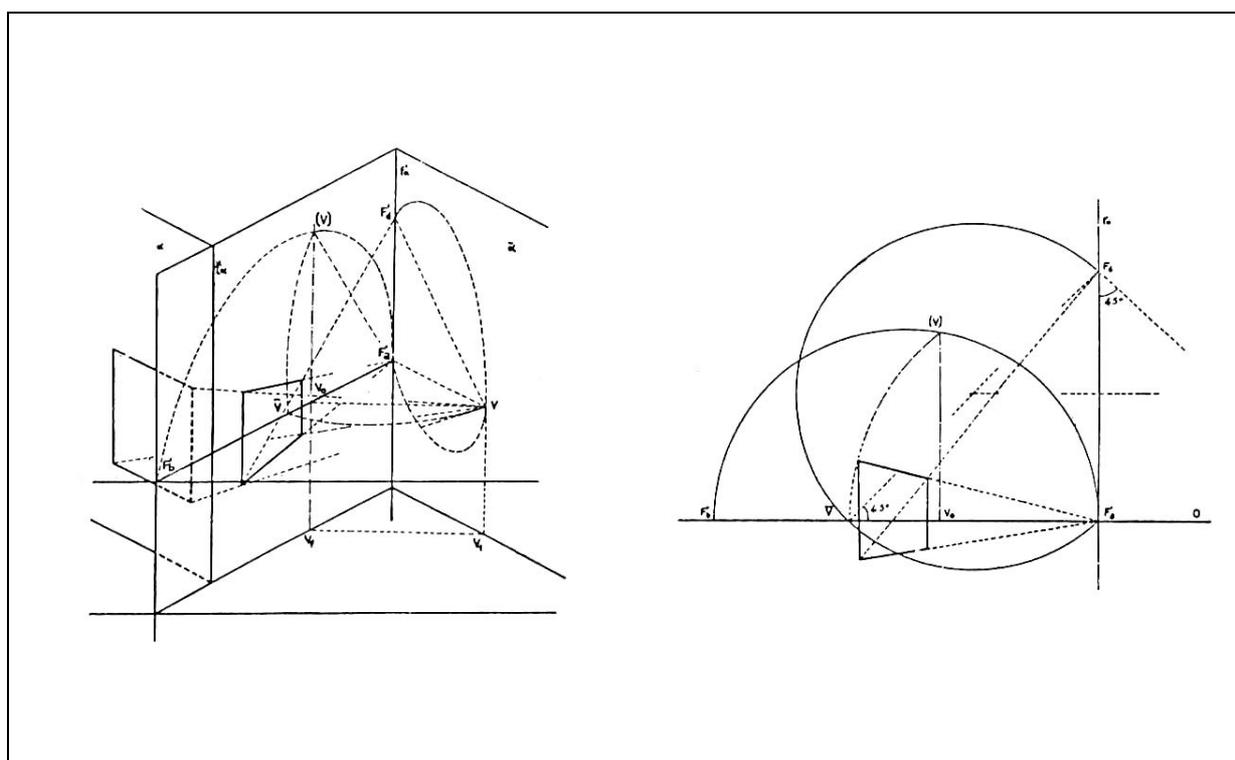
<sup>18</sup> *Ibidem*

quadrato e capace dell'angolo di  $45^\circ$  (operazione sostituibile dal tracciamento di una retta che formi un angolo di  $45^\circ$  con la retta d'orizzonte e con la retta di fuga del piano verticale contenente il quadrato); l'intersezione dell'arco con la linea d'orizzonte va ribaltata, facendo centro sul punto di fuga dei lati non verticali del quadrato, sulla semicirconferenza avente come diametro i due punti di fuga dell'angolo retto orizzontale; il punto d'incontro è il ribaltamento sul quadro del punto di vista, la sua distanza dall'orizzonte è la distanza principale.

Al procedimento precedente è possibile ritornare se sul piano verticale «è presente un elemento circolare in prospettiva accidentale, tale cioè che la relativa immagine sia un'ellisse»<sup>19</sup>, cioè «si considera il quadrato circoscritto al cerchio e avente due lati paralleli al quadro, in questo caso verticali»<sup>20</sup>. Nel caso della presenza di un rettangolo (o semicerchio, ricondotto a rettangolo circoscritto) di cui si conosca il rapporto dei lati (e quindi gli angoli con le diagonali), la costruzione è analoga a quella del quadrato, con la differenza dell'ampiezza dell'angolo (non più  $45^\circ$ ).

<sup>19</sup> *Ivi* p.32

<sup>20</sup> *Ibidem*



«Fissati gli elementi che costituiscono la prima orientazione è necessario, per completare la rappresentazione del piano geometrico, individuare la fondamentale, traccia di tale piano, di cui come si è detto la retta di orizzonte è la fuga. Poiché la distanza tra la retta di orizzonte e la fondamentale è circa uguale all'altezza dell'osservatore, se questa fosse nota si potrebbe riportare, dopo averla opportunamente ridotta in scala, sul foglio di lavoro, e quindi eseguire le operazioni di restituzione dell'immagine. Ma, non essendo generalmente conosciuta tale misura, la determinazione della fondamentale si può effettuare conoscendo la misura di un segmento orizzontale o verticale»<sup>21</sup>.

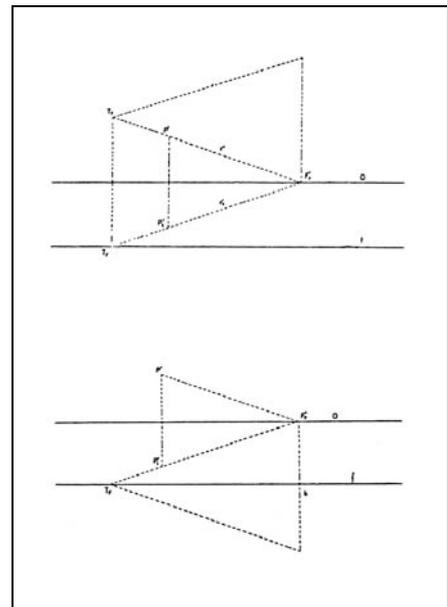
Un metodo semplice per l'individuazione della fondamentale si basa sulla conoscenza di un segmento verticale. «E' noto dal metodo delle proiezioni centrali che i segmenti che appartengono al quadro coincidono con la propria immagine: su questa proprietà è basato il procedimento mediante il quale viene costruita la prospettiva di un segmento verticale dato, avente un estremo sul geometrico, e quindi atto alla determinazione delle altezze prospettiche. In sostanza si opera una traslazione sul quadro della semiretta cui appartiene il segmento, si stacca su questa la misura del segmento oggettivo, che si riporta infine sulla semiretta-immagine»<sup>22</sup>. Nota la posizione dell'orizzonte, si disegna nella rappresentazione prospettica una qualsivoglia retta orizzontale passante per l'estremo superiore del segmento verticale, e la sua proiezione sul geometrico; dal punto di fuga comune a entrambe, si stacca un segmento di misura reale pari a quella nota del segmento prospettico, secondo l'opportuno fattore di scala; dall'estremo superiore di questo si conduce una retta parallela alla retta di fuga sul geometrico e si trova l'intersezione con la retta di fuga passante per l'estremo superiore del segmento verticale della rappresentazione prospettica; da tale punto

<sup>21</sup> *Ivi* p.35

<sup>22</sup> *Ibidem*

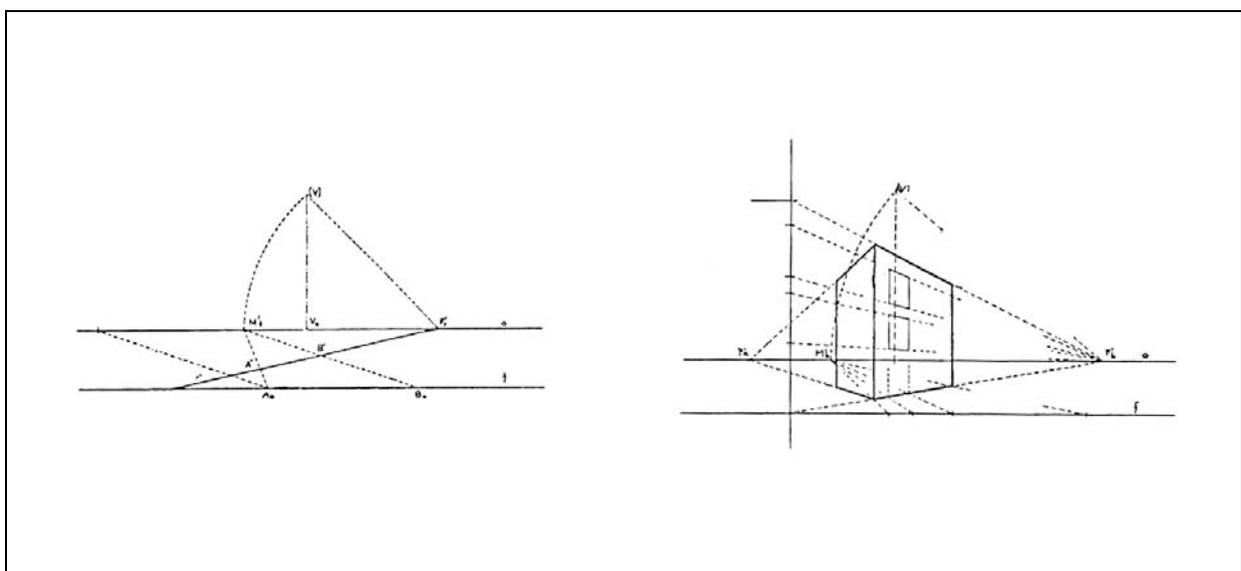
condotta una verticale, l'intersezione con questa con il prolungamento della retta di fuga sul geometrale determina la posizione della fondamentale. «La costruzione si semplifica se il segmento in vera lunghezza viene tracciato al di sotto della retta»<sup>23</sup>.

Se è nota invece la misura di un segmento del geometrale ci si può servire di un punto di misura della retta cui appartiene il segmento; facendo centro sulla fuga di tale retta, si ribalta sull'orizzonte la proiezione del centro di vista sul quadro, intercettando così il punto di misura occorrente; si tracciano dal punto di misura le rette che passano per gli estremi del segmento prospettico; a partire dal punto di misura, si stacca sull'orizzonte un segmento di dimensione reale pari a quella nota del segmento prospettico, secondo l'opportuno fattore di scala; la costruzione del parallelogramma mostrata in figura sottostante determina la posizione della retta fondamentale. «Talvolta il segmento orizzontale di misura nota, non appartiene al piano geometrale: se ne determini allora la proiezione ortogonale su tale piano, proiettandone gli estremi sulla retta del geometrale che appartiene allo stesso piano verticale del segmento dato»<sup>24</sup>, così da poter utilizzare il procedimento grafico descritto in precedenza.



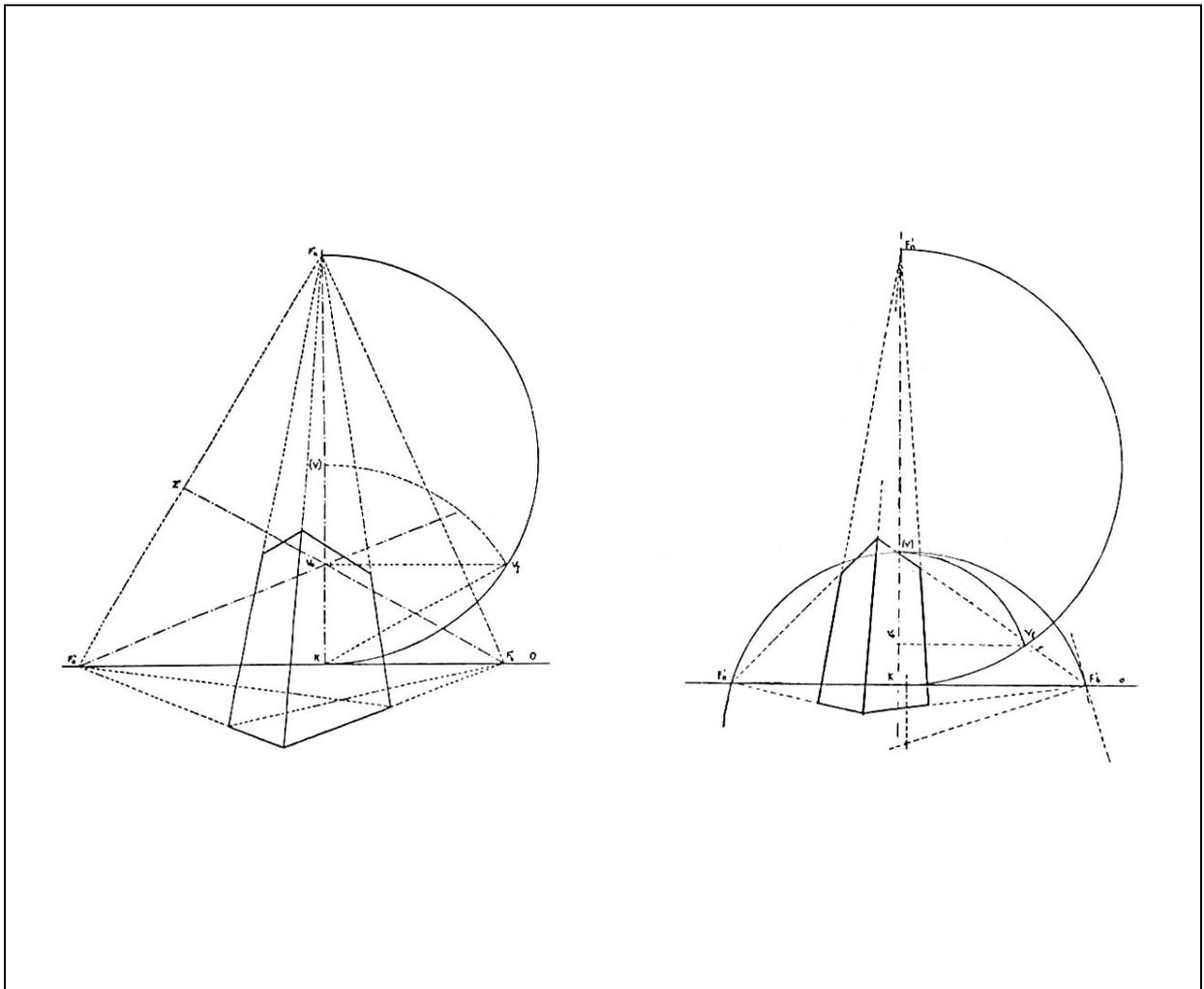
<sup>23</sup> Ivi p.37

<sup>24</sup> Ivi p.38



«Per la restituzione dei prospettivi, mediante i punti di misura si ottengono dunque sulla fondamentale le lunghezze di tutti i segmenti orizzontali; ciascuna posizione altimetrica viene poi determinata con il metodo delle altezze usato alla rovescia, vale a dire proiettando sulla traccia del piano di ciascuna faccia, le diverse altezze prospettiche e ottenendone così la vera misura. Quando si voglia restituire il contorno planimetrico è opportuno servirsi invece dell'omologia di ribaltamento, che trasforma l'immagine del contorno nella sua vera forma»<sup>25</sup>; di tale corrispondenza biunivoca, la retta fondamentale è l'asse dell'omologia, il cui centro è il punto di vista ribaltato sul quadro e una delle rette limiti è la retta di orizzonte.

<sup>25</sup> Ivi p.39



Nelle succitate *Note di Fotogrammetria Applicata all'Architettura*, Anna Sgrosso chiarifica come la restituzione prospettica sia possibile anche a partire da prospettive a quadro inclinato. «Se nell'immagine dell'edificio gli spigoli verticali non risultano paralleli, la ripresa fotografica è stata eseguita con l'asse dell'apparecchio inclinato: verso l'alto, se l'oggetto è di altezza notevolmente superiore rispetto a quella dell'osservatore; verso il basso, se l'operatore occupava una posizione più elevata rispetto all'oggetto, il piano della pellicola, sempre ortogonale all'asse, risulta dunque non più ortogonale al geometrico, cioè verticale, bensì inclinato: nel primo caso verso l'operatore, nel secondo verso l'oggetto. La prospettiva geometrica, cui tale immagine viene assimilata, dicesi perciò a quadro *inclinato*»<sup>26</sup>. In questo tipo di rappresentazione, gli spigoli verticali hanno le immagini convergenti nel punto di fuga delle rette ortogonali al geometrico. Qualora nella rappresentazione prospettica (o immagine fotografica) compaiano due piani fra loro ortogonali, determinato l'orizzonte, secondo le modalità note relative alle prospettive a quadro verticale, e individuati i due punti di fuga delle rette orizzontali appartenenti a tali piani, il punto principale è il punto d'incontro delle altezze del triangolo avente come vertici i due succitati punti di fuga e il punto di fuga delle verticali; la distanza principale è determinata da un segmento che ha un estremo sul punto principale e l'altro sulla semicirconferenza, avente come diametro il segmento ortogonale all'orizzonte, delimitato dall'orizzonte stesso e dal punto di fuga delle verticali, condotto parallelamente alla retta d'orizzonte. Se nella rappresentazione i due piani verticali non sono fra loro ortogonali, la costruzione mostrata in figura sottostante -a partire dalla semicirconferenza, avente come corda il segmento che unisce i due punti di fuga delle rette orizzontali appartenenti a tali piani, e capace dell'angolo dagli stessi formato- consente di determinare gli elementi

<sup>26</sup> Ivi p.41

di orientazione occorrenti. In una prospettiva a quadro inclinato ci si può servire di un segmento, orizzontale o verticale, di lunghezza nota, per la determinazione della fondamentale. «Supposto dapprima che sia nota la lunghezza di un segmento orizzontale ... appartenente al geometrale (oppure un qualsiasi segmento orizzontale, di cui è allora necessario determinare la proiezione geometrale, proiettandone gli estremi dell'immagine ... su una retta del geometrale), la fondamentale è la parallela alla retta» d'orizzonte, «condotta per la traccia della retta a cui appartiene il segmento orizzontale»<sup>27</sup>; la costruzione grafica che consente di determinare detta traccia è mostrata nella figura nella pagina successiva. «Qualora nel fotogramma non sia compreso o non sia leggibile il piano di calpestio e quindi il lato inferiore della faccia verticale su cui determinare la proiezione geometrale del segmento ... si può rappresentare una qualsiasi retta parallela ...(al segmento orizzontale)... preferibilmente al di sotto della retta di orizzonte, e che abbia dunque lo stesso punto di fuga. Mediante tale retta si ottiene con le costruzioni sopra descritte una fondamentale fittizia, quale traccia del piano parallelo al geometrale che contiene la retta ausiliaria: la fondamentale fittizia consente ugualmente la restituzione delle misure»<sup>28</sup>. La fondamentale fittizia è utilizzata automaticamente dal *software Brunelleschi* in tutti i casi di restituzione, come sarà chiarito successivamente (pag. 44). «Se è nota la misura di un segmento verticale ... (con un estremo ... sul geometrale, o sul piano della fondamentale fittizia), si può operare il ribaltamento sul quadro del piano di profilo che contiene il segmento»<sup>29</sup>.

Ottenuti i necessari elementi di orientazione interna ed esterna, è possibile procedere con le operazioni di restituzione in vera forma dell'immagine prospettica. «Tali operazioni, che consentono di rappresentare l'oggetto nel metodo di Monge, come si è accennato, sono le inverse a quelle necessarie alla costruzione prospettica.

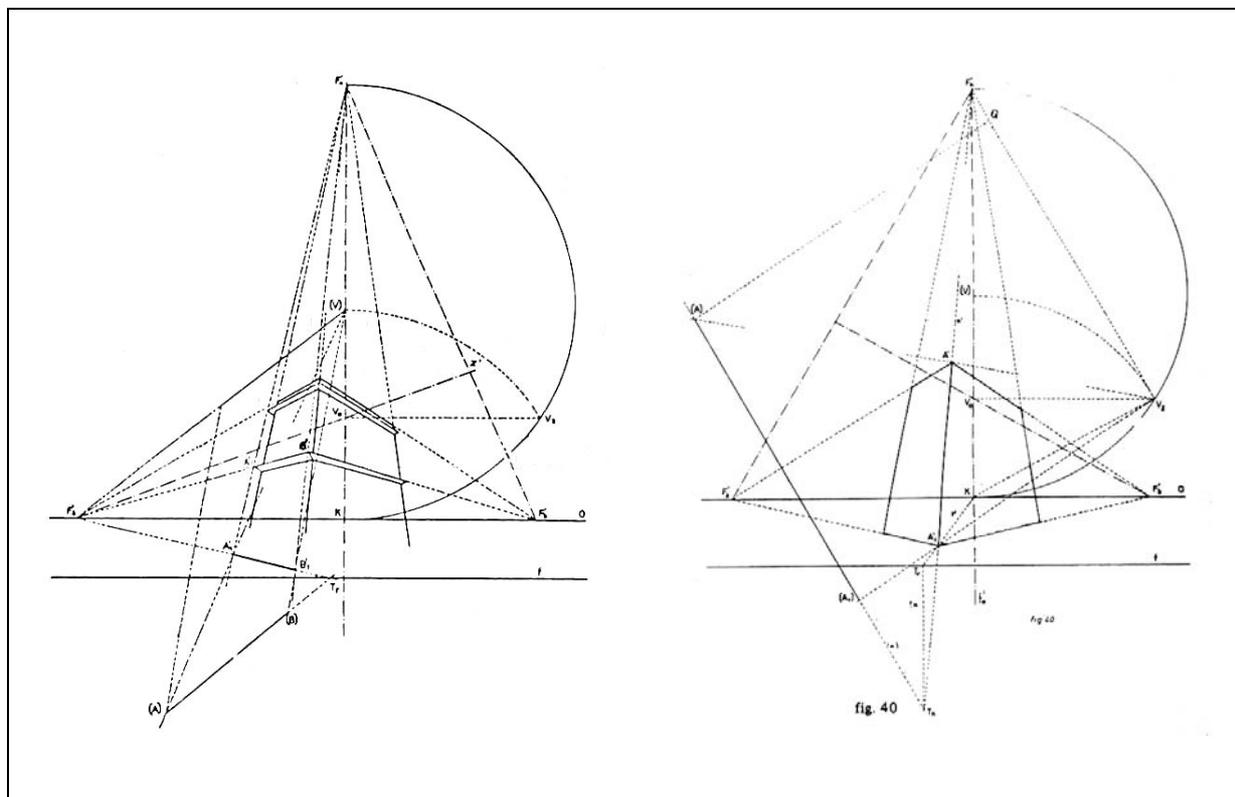
<sup>27</sup> *Ivi* p.50

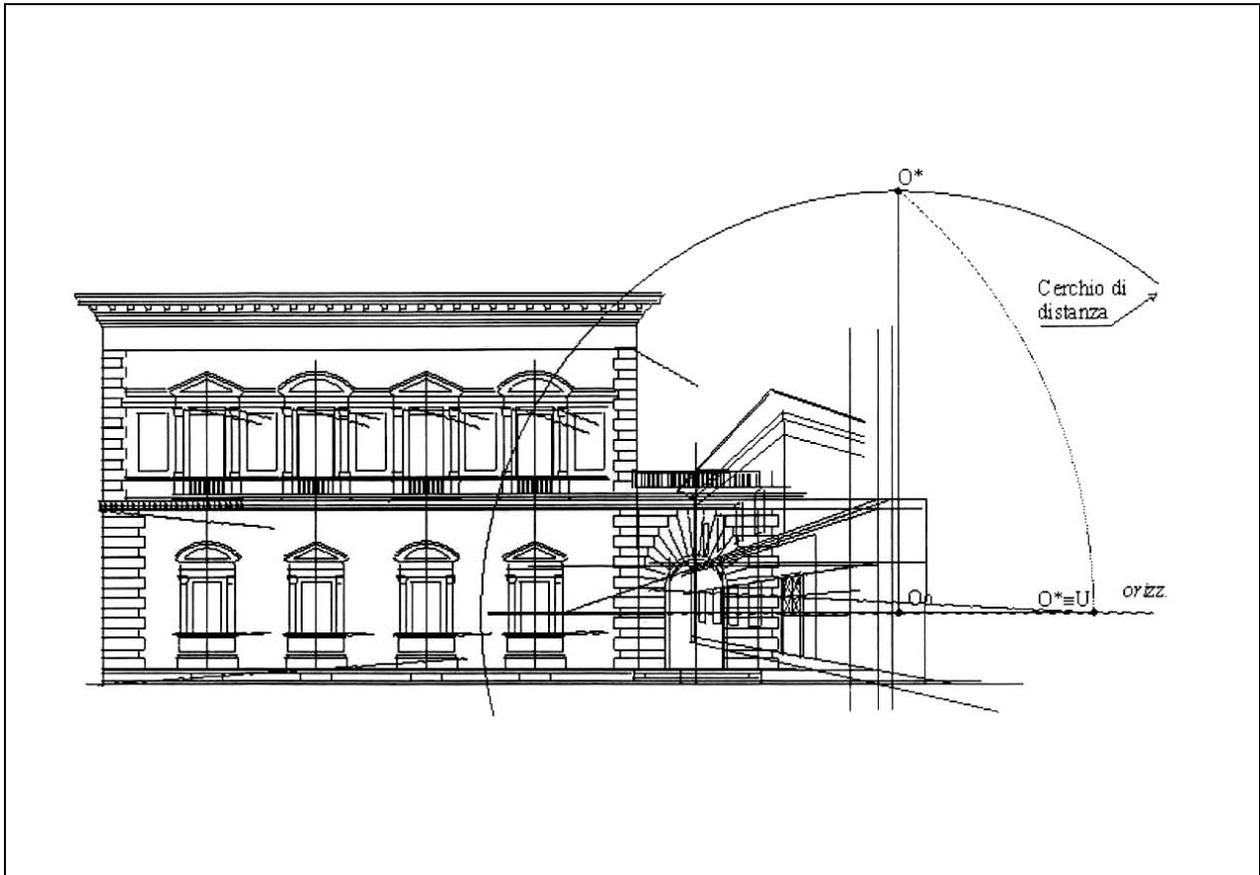
<sup>28</sup> *Ibidem*

<sup>29</sup> *Ivi* pp.51-52

Per restituire il contorno planimetrico, abbiamo visto che ci si può servire dell'*omologia di ribaltamento*, oppure del metodo dei *punti di misura*, mediante i quali si ottengono sulla fondamentale le vere lunghezze (nella stessa scala del riferimento) di tutti i segmenti orizzontali. Per restituire i prospetti, se il fotogramma è a quadro verticale, si proiettano le altezze prospettiche da un qualunque punto di fuga «della retta» di orizzonte «sul quadro, vale a dire sulla traccia del relativo piano verticale. Se il fotogramma è a quadro inclinato, le altezze prospettiche vanno proiettate,» dal centro di vista ribaltato sul quadro, «sulla ribaltata della retta cui ciascuna appartiene; per limitare il numero delle costruzioni è opportuno trasportare, mediante l'uso dei punti di fuga, le diverse altezze prospettiche (di finestre, balconi, cornici, ecc.) su uno spigolo verticale particolarmente evidente e proiettarle poi ... sulla retta ribaltata dello spigolo stesso, dove le altezze vengono fornite in vera grandezza»<sup>30</sup>.

<sup>30</sup> Ivi p.55

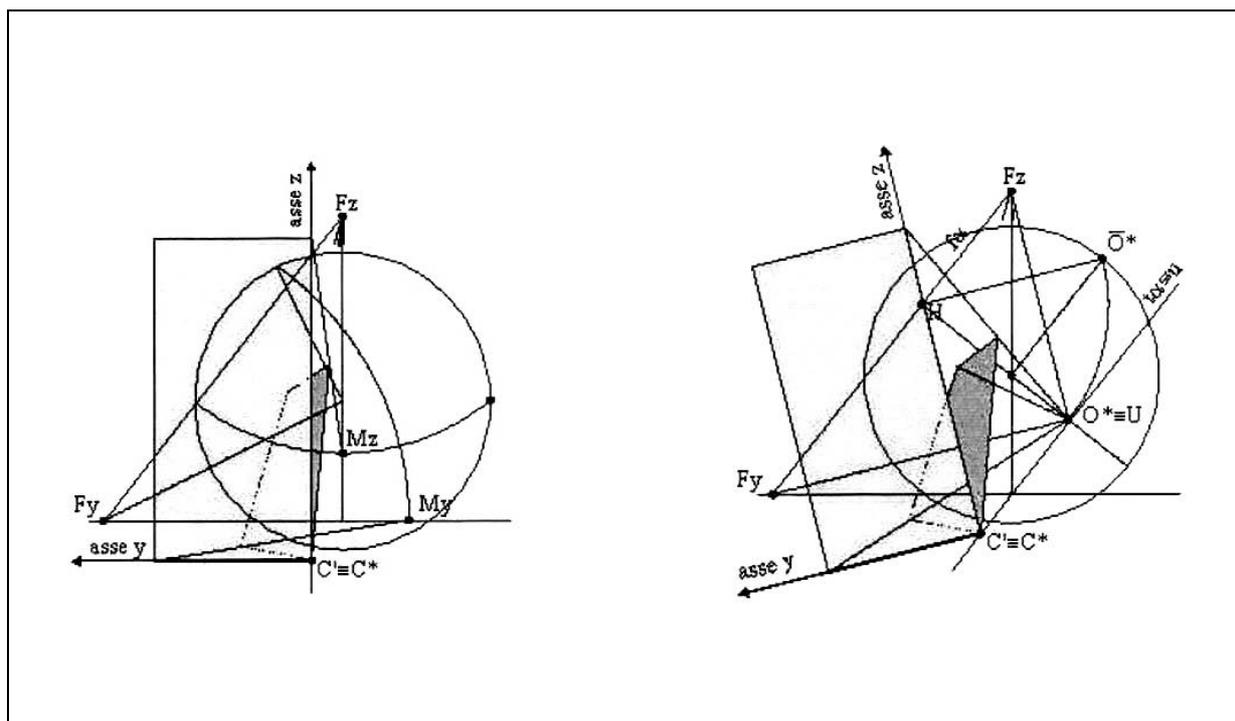




Una comune applicazione dei procedimenti di resituzione prospettica è quella denominata “raddrizzamento fotogrammetrico” dei prospetti. Tali procedimenti sono stati nel corso degli anni sviluppati all’interno di molti *software* per personal computer, sia gratuiti che commerciali, pertanto il *software Brunelleschi*, oggetto di questa tesi di dottorato, non ne costituirà ulteriore riproposizione –pur se, come sarà mostrato in seguito, consentirà di restituire alla vera forma qualsiasi elemento sintetizzato dalla scena prospettica, pertanto anche elementi di prospetto. Marco Canciani<sup>31</sup> ne illustra in maniera efficace premesse, fondamenti scientifici e una applicazione. «Nei metodi di rilievo-restituzione, che abbiano carattere speditivo, assume una certa rilevanza la restituzione metrico-dimensionale di facciate di edifici uniplanari, o con aggetti trascurabili, effettuata da singolo fotogramma. Tale processo, che può definirsi come

<sup>31</sup> Seminario di Marco Canciani sul raddrizzamento fotogrammetrico (per il corso di rilievo della professoressa Cristiana Bedoni, Università degli Studi di Roma), da cui sono tratte le figure nelle pagg.26-30

metodo di rilievo-restituzione di tipo digitale, non convenzionale, giacché non rientra nei metodi normalmente utilizzati nella Fotogrammetria, si realizza attraverso un processo sequenziale di tipo informatico, composto di tre fasi distinte: 1) l'acquisizione dell'immagine fotografica al computer tramite scansione e la conversione dell'immagine fotografica in una matrice digitale di pixel ...; 2) la trasformazione numerica di ogni singolo punto, dalla sua collocazione nello spazio proiettato sulla foto, ad una collocazione bidimensionale in proiezione ortogonale, definito come raddrizzamento dell'immagine; 3) la fase di elaborazione grafica sovrimposta alla immagine raddrizzata (overmapping), seguente al processo di acquisizione-trasformazione-conversione dei dati digitali, che produce una restituzione indiretta delle misure e delle caratteristiche formali della facciata fotografata. Tale procedimento utilizza delle prese attraverso comuni apparecchi fotografici, con un inclinazione sull'asse  $x$  e  $y$ , verticale ed orizzontale, libera, e con rotazione teorica limitata sull'asse focale  $z$ , perpendicolare al fotogramma».

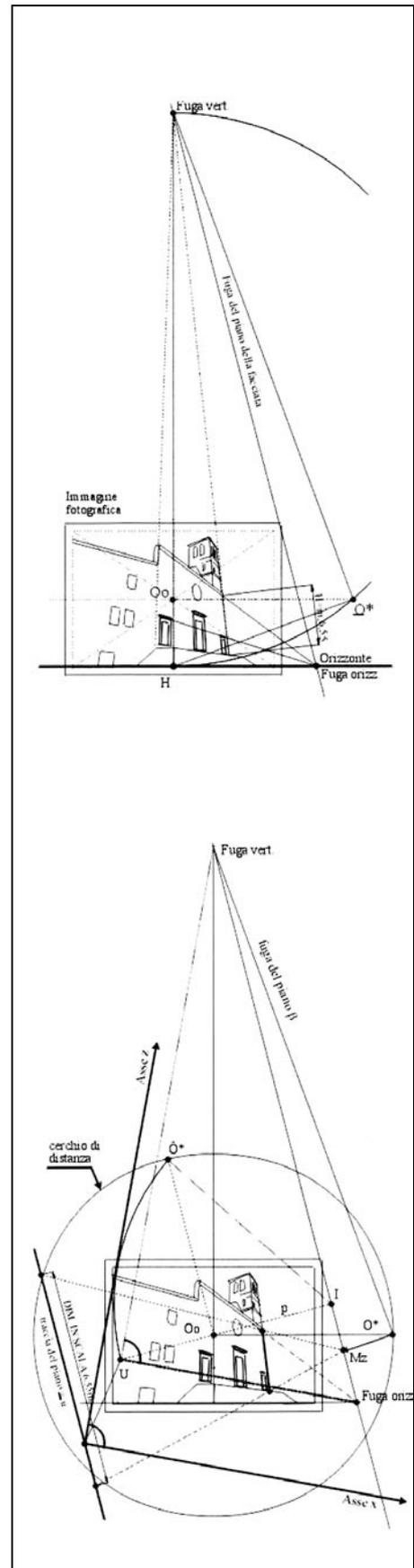


Come già scritto in precedenza, «la costruzione geometrica, associata alla presa fotografica, è, come è noto, riconducibile in linea teorica, a meno delle deformazioni dovute all'ottica della camera, alle regole e ai metodi della geometria proiettiva, in particolare alla cosiddetta prospettiva rovescia, con il centro di proiezione interposto tra il quadro e l'oggetto proiettato». Pertanto, è possibile determinare gli elementi dell'orientamento interno. «La trasformazione, dal punto di vista analitico, avviene secondo una trasposizione da una matrice  $x',y'$  originaria, riferita all'immagine, ad una matrice di destinazione  $x,y$ , corrispondente alla prima, dove sono rappresentati i punti nella loro collocazione reale, in proiezione ortogonale e non più centrale, secondo le seguenti equazioni:  $intersezx = (fuvx * mvert - fuvy + cost3) / (mvert - morizz)$ ;  $interseyz = mvert * (intersezx - fuvx) + fuvy$ ; la costruzione geometrica, che si riferisce alle regole della geometria descrittiva, a cui la restituzione da singolo fotogramma fa riferimento, viene eseguita per mezzo di un cosiddetto box di misurazione, che è definito secondo un sistema di piani limite misuratori, associati a quello della facciata: un piano inferiore orizzontale, ed uno laterale proiettante, che taglia la facciata in esame sul bordo laterale. Nella restituzione da proiezione a quadro inclinato, come è quella fotografica, si adotta solitamente il *metodo risolutivo del ribaltamento* del piano della facciata, in modo da utilizzare la relazione omologica che intercorre tra gli elementi disposti sul piano medesimo. Tale costruzione, che viene normalmente adottata nel procedimento di restituzione ... prevede l'individuazione degli elementi dell'omologia, e cioè l'asse, corrispondente alla traccia del piano da raddrizzare, il centro  $O^*$  ribaltato sul quadro, e una coppia di punti corrispondenti, posti rispettivamente sui due piani». ... «Al fine di ridurre le elaborazioni al computer e semplificare l'algoritmo di calcolo, si è preferito utilizzare il box di misurazione che, per la sua disposizione geometrica relazionata

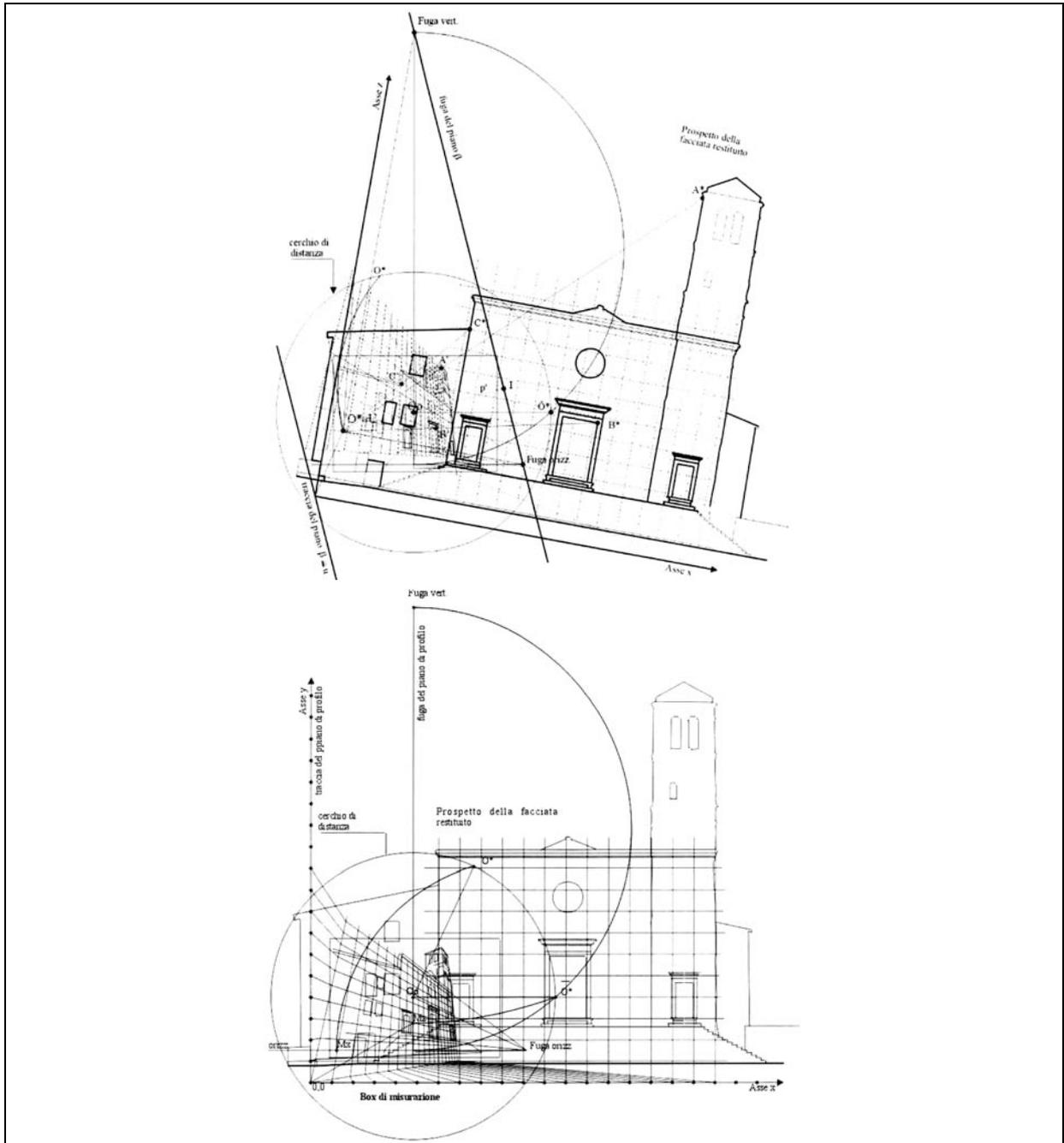
direttamente all'orientamento interno della camera, e svincolato dalla posizione del piano di facciata, produce direttamente un sistema di assi ortogonali  $x$  e  $y$  sul piano di restituzione, già orientato secondo ascisse disposte in orizzontale, e ordinate disposte in verticale. ... Gli assi  $x$  ed  $y$  rappresentano, ... le rispettive intersezioni, in quanto linee di traccia, dei piani fondamentali di restituzione; il primo taglia in senso orizzontale il piano di facciata ... il secondo, posto verticalmente, seziona il piano di facciata lateralmente ed essendo un piano proiettante, produce, come è noto, una retta intersezione, perpendicolare all'orizzonte. Tali intersezioni ... vengono misurate attraverso una serie di proiezioni delle unità metriche dai rispettivi punti di misura  $M_x$  ed  $M_y$ » (figura a pag. 27).

«Sull'immagine ... ottenuta è possibile misurare alcune dimensioni relative agli oggetti posizionati sul piano di facciata, dopo aver calibrato la scala di restituzione attraverso l'immissione dei dati di una lunghezza reale». La considerazione successiva è un monito da tenere ben presente quando si eseguono operazioni di restituzione prospettica –raccomandazione che pur riferendosi ad essa qualora assistita al computer, ancor più è valida quando eseguita manualmente su foglio di carta- «dato che il raddrizzamento si basa su diversi parametri, quali la posizione di presa, la qualità dell'immagine originaria e la definizione dell'immagine risultata dal raddrizzamento, si possono determinare risultati considerevolmente differenti, derivati da variazioni minime nel sistema di calcolo».

L'esempio di raddrizzamento proposto riguarda la facciata della chiesa di San Giuliano a Faleria; in figura sovrastante, le costruzioni geometriche per determinare gli elementi di orientamento interno del fotogramma. Determinati l'orizzonte e la fuga verticale  $F_v$ , e disegnata la semicirconfenza avente come diametro il segmento definito dalla fuga verticale e dalla sua proiezione ortogonale sulla retta d'orizzonte  $H$ , dal centro della



presa nel fotogramma "Oo mandiamo una parallela all'orizzonte sino ad incontrare la semicirconferenza nel punto O\*, che esprime il ribaltamento laterale del centro di proiezione. (Il triangolo H-O\*-Fv visualizza l'angolo con il quale è stata effettuata la ripresa). Con centro Oo e raggio Oo-O\*, possiamo ora tracciare il cerchio di distanza".



Un approccio differente al tema della restituzione prospettica, con l'utilizzo del personal computer, è quello proposto da Camillo Trevisan<sup>32</sup>, il quale è autore della predisposizione del *software Euclid* per la restituzione prospettica, da egli stesso descritto in questi termini: «Il programma *Euclid* è in grado di effettuare la restituzione prospettica per qualsiasi tipo di prospettiva a centro di proiezione unico, a quadro piano o curvo, proiettata su sfere o cilindri, solida o scenografica, purché contenente solidi o figure piane geometricamente riconoscibili. Il metodo proposto è vicino a quello tradizionale: la differenza sostanziale consiste nel fatto che, mentre per via grafica si è costretti a definire fin dall'inizio gli elementi dell'orientamento interno ed esterno della prospettiva e restituirla in funzione di questi, nel nostro caso questi parametri rimangono variabili e vengono costantemente aggiornati affinché tutte le condizioni poste siano valide, compresa ovviamente la coincidenza della prospettiva data con quella prodotta. Il programma si basa dunque sulla continua verifica di leggi o regole compositive imposte ad un modello tridimensionale in costante deformazione. Se e quando tutte le norme sono rispettate ... il modello reale così trovato rappresenterà la restituzione della prospettiva data». Le basi teoriche di fondamento del *software Euclid*, valide anche per il *software Brunelleschi* -tema di questa tesi di dottorato- sono definite dall'autore con chiarezza nelle premesse della sua trattazione; egli esprime in questi termini i concetti di "modello reale" e "modello prospettico": «definiamo ... come modello reale l'insieme dei punti che individuano i vertici dei solidi rappresentati nella prospettiva: può quindi essere visto in pianta o in alzato o in qualsiasi altra forma rappresentativa voluta. Costituisce in pratica il modello utilizzato dall'artista per produrre la prospettiva ed è, per definizione, conosciuto topologicamente ma sconosciuto o meglio non perfettamente individuato nei valori delle coordinate: sarà compito del programma

<sup>32</sup> Professore Associato in Disegno (IUAV) dal 2002, autore della predisposizione del programma *EUCLID* per la restituzione prospettica (1990-2000). "Il programma è in grado di effettuare la restituzione prospettica per qualsiasi tipo di prospettiva a Centro di Proiezione unico, a Quadro piano o curvo, proiettata su sfere o cilindri, "solida" o scenografica, purché contenente solidi o figure riconoscibili nella loro configurazione piana o tridimensionale"; autore delle figure a pag.34.

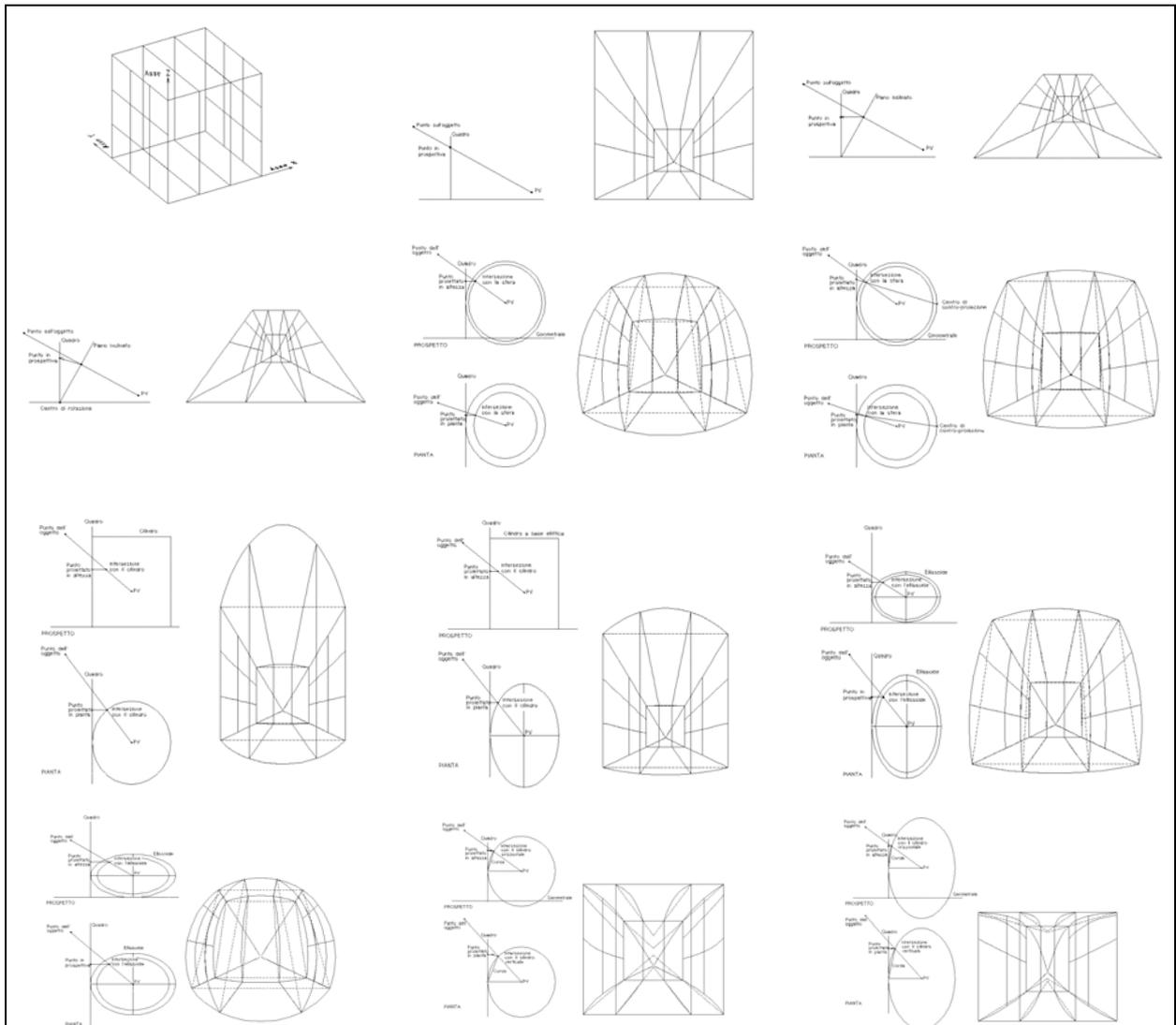
definirlo spazialmente in tutte le sue articolazioni e verificare che la topologia presunta sia corretta. Per modello prospettico intenderemo al contrario l'insieme dei punti del modello reale messi in prospettiva dal programma *Euclid*, utilizzando il punto di osservazione scelto dal programma stesso o dall'utente: tale modello deve, in linea teorica, coincidere con la prospettiva data: con questo nome chiameremo infatti l'insieme dei punti (omologhi a quelli del modello reale e prospettico) digitalizzati sulla prospettiva o ricavati in qualsiasi altro modo».

Il metodo di lavoro del *software Euclid* viene messo a confronto con i metodi tradizionali per metterne in evidenza le diverse potenzialità; «generalmente la ricostruzione della forma e dimensioni reali di uno o più oggetti rappresentati in prospettiva, viene intesa come l'operazione inversa della prospettiva stessa: un metodo che permette di ricavare, fatte alcune supposizioni sulla forma reale di un oggetto, gli elementi dell'orientamento interno ed esterno della prospettiva. Tali parametri si applicheranno anche a tutti gli altri oggetti rappresentati. È ovvio che tale metodo può essere applicato con successo se gli oggetti raffigurati sono ben individuabili e riconoscibili. ... Può accadere però che la restituzione, imperniata sulle presunte caratteristiche di un oggetto, non si attagli bene al resto della scena visualizzata, ad altri solidi dislocati in posizioni diverse: sarà pertanto necessario riformulare nuove idee, ripetere il procedimento di restituzione e controllo fino a quando tutti gli oggetti risulteranno credibili. Vale a dire che il classico metodo di restituzione, basandosi su supposizioni spesso non dimostrabili a priori, diventa, da esatto, quasi sempre ricorsivo. ... Per finire lo studioso si trova a volte a dover restituire scene messe in prospettiva non classica (legittima), ma ad esempio proiettate su volte o cupole, oppure sferiche o cilindriche o anamorfiche o addirittura solide: in questi casi i problemi si complicano

ulteriormente rendendo veramente ardua la soluzione con metodi tradizionali. Il metodo proposto, prescindendo dalla tecnica utilizzata per la costruzione della prospettiva, risulta applicabile virtualmente in tutti i casi, anche i più complessi e di più difficile soluzione».

Le tecniche di rappresentazione prospettica vengono distinte in quattro categorie principali: «Prospettiva centrale su superficie piana a quadro piano. Fanno parte di questa categoria sia gli infiniti esempi di pittura dal Rinascimento in poi, sia le anamorfose piane ..., ed in genere tutte le prospettive, sia a quadro verticale che inclinato». «Prospettiva centrale su superficie non piana. È questo il caso di scene proiettate su emisfere (cupole) o semi cilindri (volte a botte) o altre superfici non piane (volte a crociera, cupole paraboloidi o ellissoidi). In questo caso la scena, messa in prospettiva tradizionale su di un cartone piano, è stata proiettata da un punto, generalmente non al centro della sfera o sull'asse del cilindro, sulla cupola o volta o sulla superficie: solo un osservatore posto in quel punto percepirà una vista corretta in tutte le sue parti; da qualsiasi altro punto gli oggetti rappresentati appariranno via via più deformi ed assurdi quanto più l'occhio sarà distante dal corretto centro di proiezione e di vista». «Prospettiva su superficie piana a quadro non piano. Le proiettanti vengono intercettate da una superficie non piana (quadro principale), ed i punti così definiti successivamente contro-proiettati o sviluppati su piano (quadro secondario)». «Prospettive solide o in rilievo. Con questi termini si intendono architetture o scenografie deliberatamente costruite come *trompe* per dare all'osservatore l'impressione di vedere solidi di forme e dimensioni diverse da quelle reali. Sono assimilabili a questa categoria, ad esempio, il falso coro di S. Satiro, ...». Il metodo di restituzione proposto da Camillo Trevisan è dallo stesso studioso definito "pseudo empirico"; esso viene messo in pratica «non ricostruendo ma costruendo

la prospettiva e confrontandola poi con l'originale e verificando contemporaneamente tutte le supposizioni fatte circa tutti gli oggetti presi in considerazione. Tale procedimento pseudo empirico di: supposizione della realtà - calcolo della prospettiva - confronto con l'originale, evidentemente non può essere eseguito con metodi grafici manuali, già di per sé lunghi e approssimativi. In questo caso l'uso dell'elaboratore elettronico garantisce sia la velocità che la precisione e risulta in grado di operare indifferentemente con tutti i tipi di prospettiva proprio perché ne utilizza la caratteristica comune appena considerata».



## LA RESTITUZIONE PROSPETTICA CON *BRUNELLESCHI*, POTENZIALITÀ E LIMITI

*Brunelleschi* è programmato per la restituzione di prospettive a quadro verticale, siano esse centrali o accidentali, in cui, per usare la terminologia dei moderni *software cad*, il punto di vista è alla stessa altezza del "punto di mira". Questa limitazione mi ha consentito di concentrare lo sforzo di programmazione sulla "struttura" fondamentale del *software*, sul suo *kernel*, nucleo portante, senza in alcun modo pregiudicare, grazie alla versatilità del linguaggio di sviluppo scelto, la possibilità di superare tale limitazione in futuro ed estendere la gamma di rappresentazioni prospettiche ammissibili alle prospettive a quadro inclinato.

La restituzione prospettica classica consiste in una serie di procedimenti grafici che consentono di desumere, da una immagine fotografica priva di distorsioni o da una rappresentazione in prospettiva correttamente eseguita, le proiezioni di Monge e quindi le dimensioni -o quantomeno le proporzioni- reali di quanto raffigurato.

La codificazione di detti procedimenti grafici all'interno *Brunelleschi* permette di facilitare enormemente e di aumentare l'accuratezza della restituzione prospettica. Come intuibile, l'immagine dovrà essere disponibile in formato digitale, secondo gli standard più comuni nell'ambito delle immagini di tipo *raster*, per consentire a *Brunelleschi* di importarla e visualizzarla sullo schermo. L'operatore dovrà avere la capacità di sintetizzare ed individuare nella prospettiva gli elementi di orientamento e geometrici; il *software* si occuperà in tempo reale di tradurre questi ultimi dalla deformazione della rappresentazione prospettica alla realtà metrica e proporzionale in un modello a tre dimensioni; come noto, da un modello tridimensionale digitale sono immediatamente desumibili le proiezioni di Monge, o assonometriche, o qualsivoglia proiezione centrale, e tutto

ciò è possibile con lo stesso *software Brunelleschi* o trasferendo il modello tridimensionale digitale in un altro *software* per successive elaborazioni.

Con *Brunelleschi* è possibile individuare sulla prospettiva gli elementi di orientamento secondo i casi noti più comuni; per facilitare il procedimento di sintesi geometrica dell'operatore è inoltre possibile, con un singolo comando, individuare sulla stessa prospettiva, oltrechè linee e piani, volumi prismatici e cilindrici, colonnati e pavimentazioni, archi e volte a botte, padiglione e crociera, ed ottenerne l'immediata restituzione nel modello.

## DETERMINAZIONE DELL'ORIENTAMENTO DELLA RAPPRESENTAZIONE PROSPETTICA

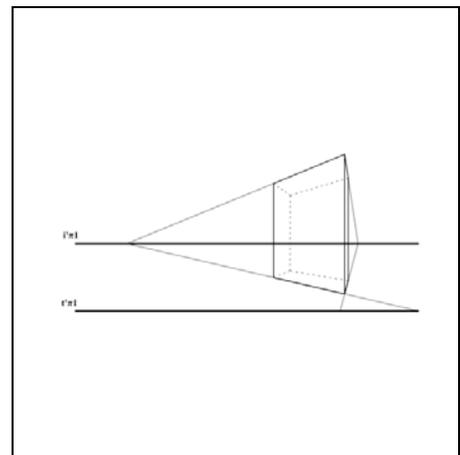
Per poter effettuare la trasformazione omologica della rappresentazione di un insieme di oggetti da proiezione centrale a proiezioni di Monge, è necessario determinare preliminarmente gli elementi di orientamento della prospettiva. E' comune la distinzione fra elementi di "orientamento interno", consistenti nella retta di orizzonte, oltrechè nei punti principale e di vista, che determinano la distanza principale, ed "orientamento esterno", ossia individuazione della retta fondamentale.

Di seguito vengono puntualmente esaminati, fra i metodi pratici di determinazione degli elementi di orientamento, quelli che sono stati messi a disposizione dell'utente finale in questa *release* del *software* di restituzione *Brunelleschi*, con alcuni cenni alle soluzioni di geometria analitica incluse negli algoritmi di calcolo codificati nel *software*.

### DETERMINAZIONE DELL'ORIENTAMENTO INTERNO

#### *Determinazione della retta d'orizzonte*

La retta d'orizzonte è la retta di fuga prospettica di tutti i piani paralleli al geometrico –piani orizzontali- ed è pertanto il luogo dei punti di fuga di tutte le rette appartenenti a tali piani. Detto ciò, è intuitivo che per determinarne la posizione in una rappresentazione prospettica, quando non evidente, è sufficiente individuare due coppie di rette parallele "orizzontali", e le loro intersezioni, come nell'esempio mostrato in figura. Nella pratica del disegno manuale, tali intersezioni possono talvolta superare i margini del foglio e costringere a procedimenti alternativi, i quali risultano però superflui sul "foglio" virtualmente infinito di uno strumento *software*.

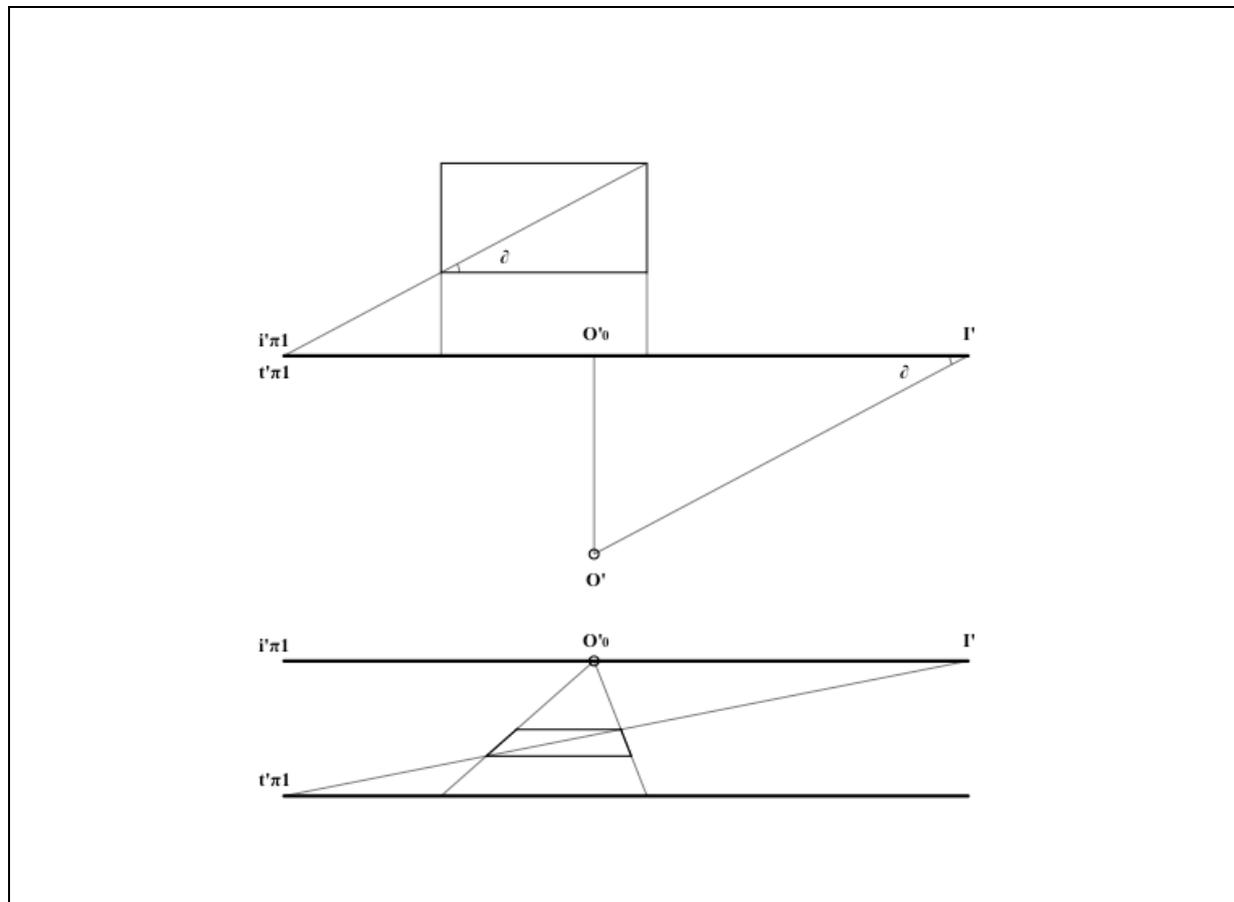


**Determinazione del punto di vista e del punto principale, dato un rettangolo di proporzioni note e con due lati paralleli al quadro<sup>33</sup>**

<sup>33</sup> La traduzione di questo procedimento grafico in forma analitica, necessaria per la codifica nel software di restituzione, parte dalla definizione, sul piano cartesiano della rappresentazione prospettica, delle coordinate dei punti del rettangolo; le equazioni delle rette a cui appartengono un lato ortogonale al quadro e una diagonale, sono ricavate dalla formula  $(x-x_1)/(x_2-x_1)=(y-y_1)/(y_2-y_1)$ , e risolte nella forma  $y=mx+q$ , facendo poi sistema con l'equazione della retta d'orizzonte  $y=i'\pi_1$  per trovare  $O'_0$  e  $I'$ . La distanza principale  $O'_0-O'$  e quindi la posizione del punto di vista sono determinati da  $(O'_y-O'_{0y})=(I'_x-O'_{0x})*\tan(\vartheta)$ .

Dato il rettangolo rappresentato in prospettiva nella parte inferiore della figura sottostante, è sufficiente individuare sull'orizzonte il punto principale  $O'_0$  e il punto di fuga della diagonale  $I'$ .

Note le proporzioni reali del rettangolo, e quindi l'angolo  $\vartheta$ , occorre tracciare, come mostrato in figura nella parte superiore, una retta a partire dal punto  $I'$ , inclinata di  $\vartheta$  rispetto alla fondamentale, e una retta ortogonale alla fondamentale a partire dal punto  $O'_0$ ; l'intersezione fra le due rette determinerà il punto di vista  $O'$  e la distanza principale  $O'_0-O'$ .

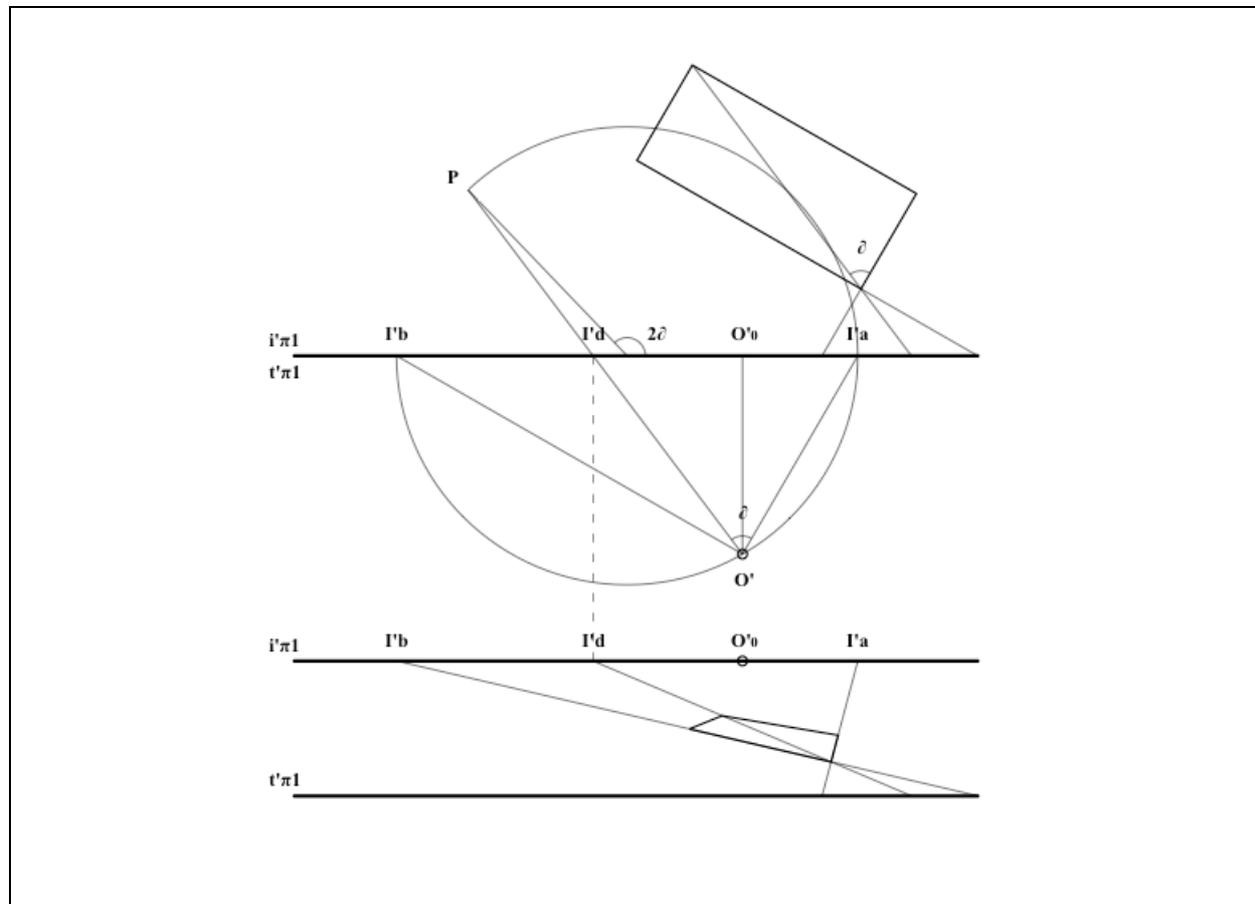


*Determinazione del punto di vista e del punto principale, dato un rettangolo di proporzioni note con lati non paralleli al quadro<sup>34</sup>*

Dato il rettangolo rappresentato in prospettiva nella parte inferiore della figura sottostante, occorre individuare sull'orizzonte i punti di fuga  $I'a$  e  $I'b$  dei lati fra loro ortogonali e il punto di fuga della diagonale  $I'd$ .

Tracciato un cerchio di diametro  $I'a-I'b$ , come mostrato in figura nella parte superiore, essendo note le proporzioni reali del rettangolo, e quindi l'angolo  $\vartheta$ , occorre staccare, su tale cerchio, un angolo di ampiezza  $2\vartheta$  e proiettare da  $I'd$  il punto  $P$  così individuato sulla circonferenza, intersecando la stessa circonferenza nel punto cercato  $O'$ ; come noto, essendo l'angolo al centro di ampiezza  $2\vartheta$ , deve essere  $\vartheta$  l'angolo alla circonferenza che insiste sullo stesso arco.

<sup>34</sup> Il punto medio del segmento  $I'a-I'b$  viene assunto come centro del piano cartesiano; l'equazione della circonferenza è  $x^2+y^2=r^2$ , con  $r=(I'b_x-I'a_x)/2$ ; il punto  $P$  ha coordinate  $\cos(2\vartheta)*r$  e  $\sin(2\vartheta)*r$ , e la retta che passa per  $P$  e  $I'd$ , risolta nella forma  $y=mx+q$ , viene posta a sistema con l'equazione della circonferenza per determinare le coordinate di  $O'$ .

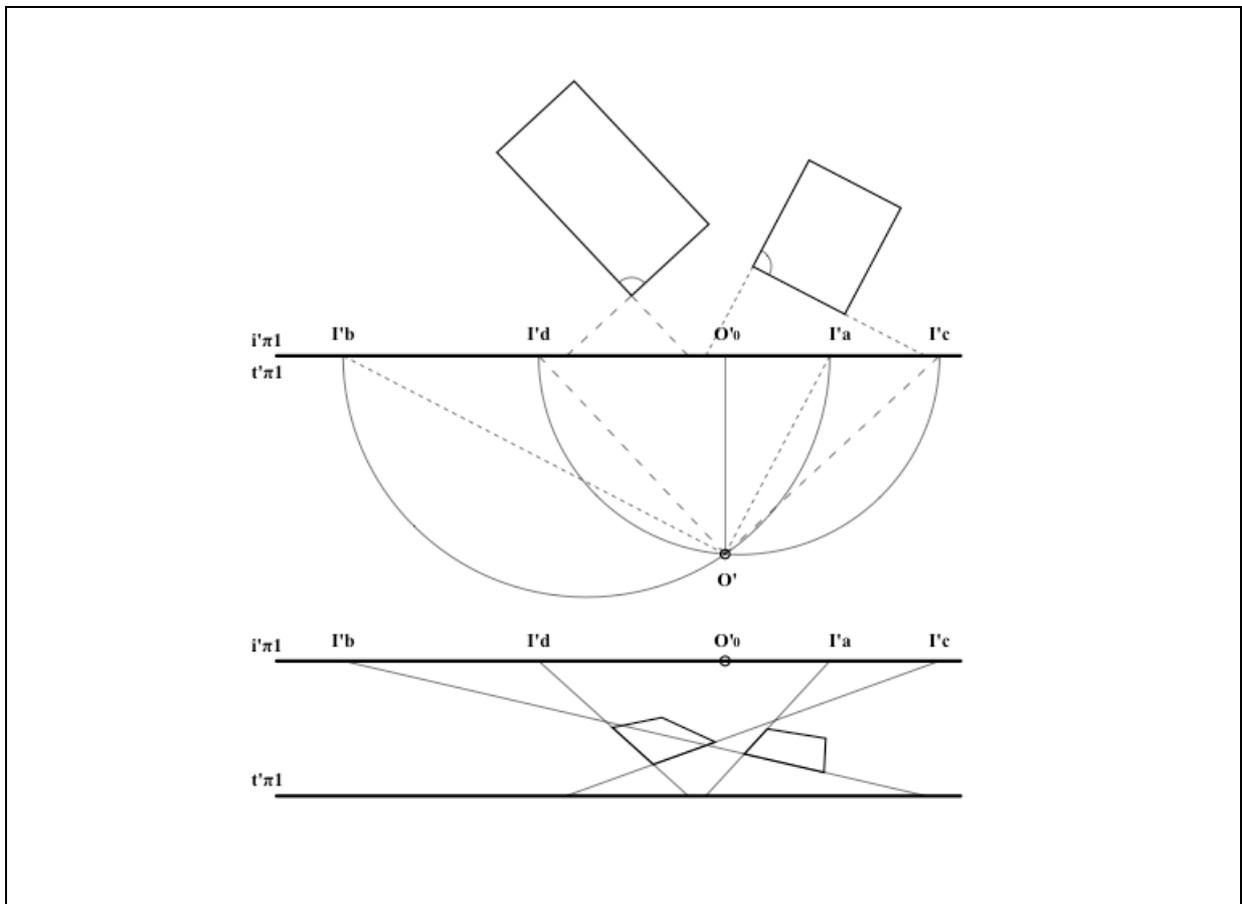


*Determinazione del punto di vista e del punto principale, dati due rettangoli diversamente orientati, con lati non paralleli al quadro<sup>35</sup>*

Dati i rettangoli rappresentati in prospettiva nella parte inferiore della figura sottostante, occorre individuare sull'orizzonte i rispettivi punti di fuga l'a, l'b, l'c, l'd.

Tracciati due semicerchi, rispettivamente di diametro l'a-l'b e l'c-l'd, come mostrato in figura nella parte superiore, la loro intersezione determina univocamente la posizione di O'.

<sup>35</sup> Il punto medio del segmento l'a-l'b viene assunto come centro del piano cartesiano; l'equazione della circonferenza di diametro l'a-l'b è  $x^2+y^2=r_{12}^2$ , con  $r_{12}=(l'b_x-l'a_x)/2$ ; sia C il centro di suddetta circonferenza e punto medio di l'a-l'b; l'equazione della circonferenza di diametro l'c-l'd, nel medesimo piano cartesiano di riferimento, può essere risolta nella forma seguente,  $(x-C_x)^2+y^2=r_{34}^2$ , con  $r_{34}=(l'd_x-l'c_x)/2$ ; le due equazioni vengono poste a sistema per determinare le coordinate di O' (l'ascissa è univoca, mentre per l'ordinata viene assunto il valore positivo).

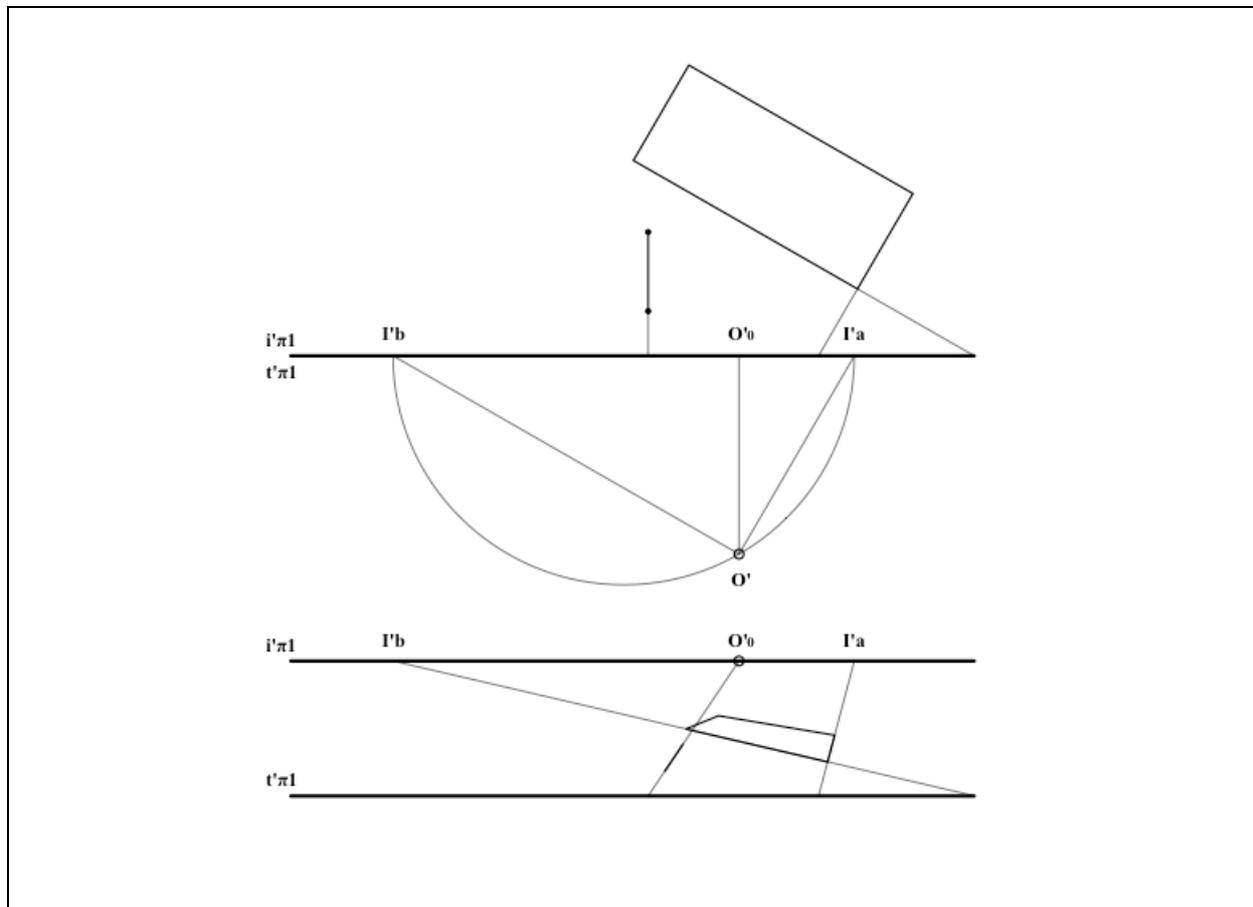


***Determinazione del punto di vista e del punto principale, dati un rettangolo con lati non paralleli al quadro e un segmento orizzontale ortogonale al quadro<sup>36</sup>***

Dato il rettangolo rappresentato in prospettiva nella parte inferiore della figura sottostante, occorre individuare sull'orizzonte i punti di fuga  $I'a$  e  $I'b$ ; il segmento orizzontale, essendo ortogonale al quadro, va in fuga nel punto principale  $O'_0$ .

Tracciato un semicerchio, di diametro  $I'a-I'b$ , come mostrato in figura nella parte superiore, si conduce a partire da  $O'_0$  una retta perpendicolare alla fondamentale, che interseca il suddetto semicerchio nel punto di vista  $O'$ .

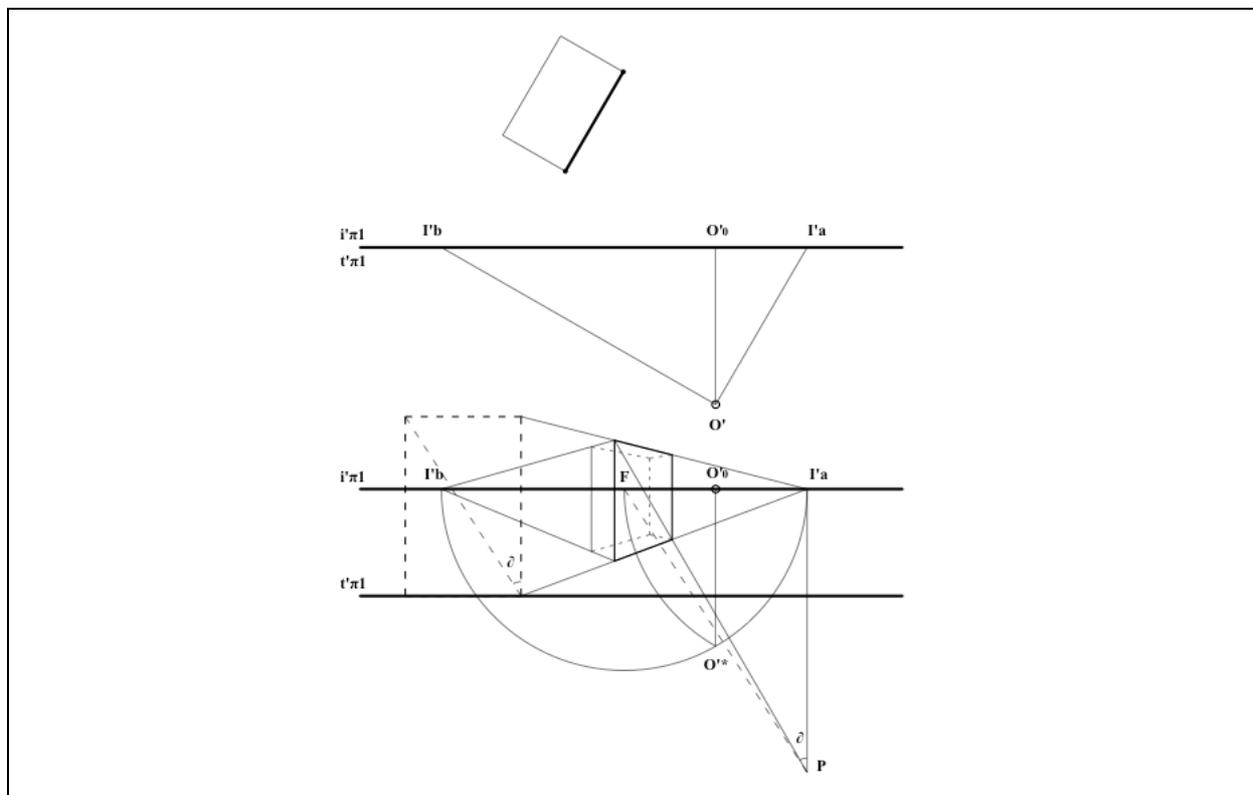
<sup>36</sup> Il punto medio del segmento  $I'a-I'b$  viene assunto come centro del piano cartesiano; l'equazione della circonferenza di diametro  $I'a-I'b$  è  $x^2+y^2=r_{12}^2$ , con  $r_{12}=(I'b_x-I'a_x)/2$ ; sia  $C$  il centro di suddetta circonferenza e punto medio di  $I'a-I'b$ ; l'equazione della retta ortogonale alla fondamentale e condotta a partire dal punto  $O'_0$ , nel medesimo piano cartesiano di riferimento, è  $x=O'_0x-C_x$ ; le due equazioni vengono poste a sistema per determinare le coordinate di  $O'$  (per l'ordinata viene assunto il valore positivo).





*Determinazione del punto di vista e del punto principale, dati un rettangolo di proporzioni note, ortogonale al piano orizzontale, e un segmento orizzontale ortogonale ad esso*

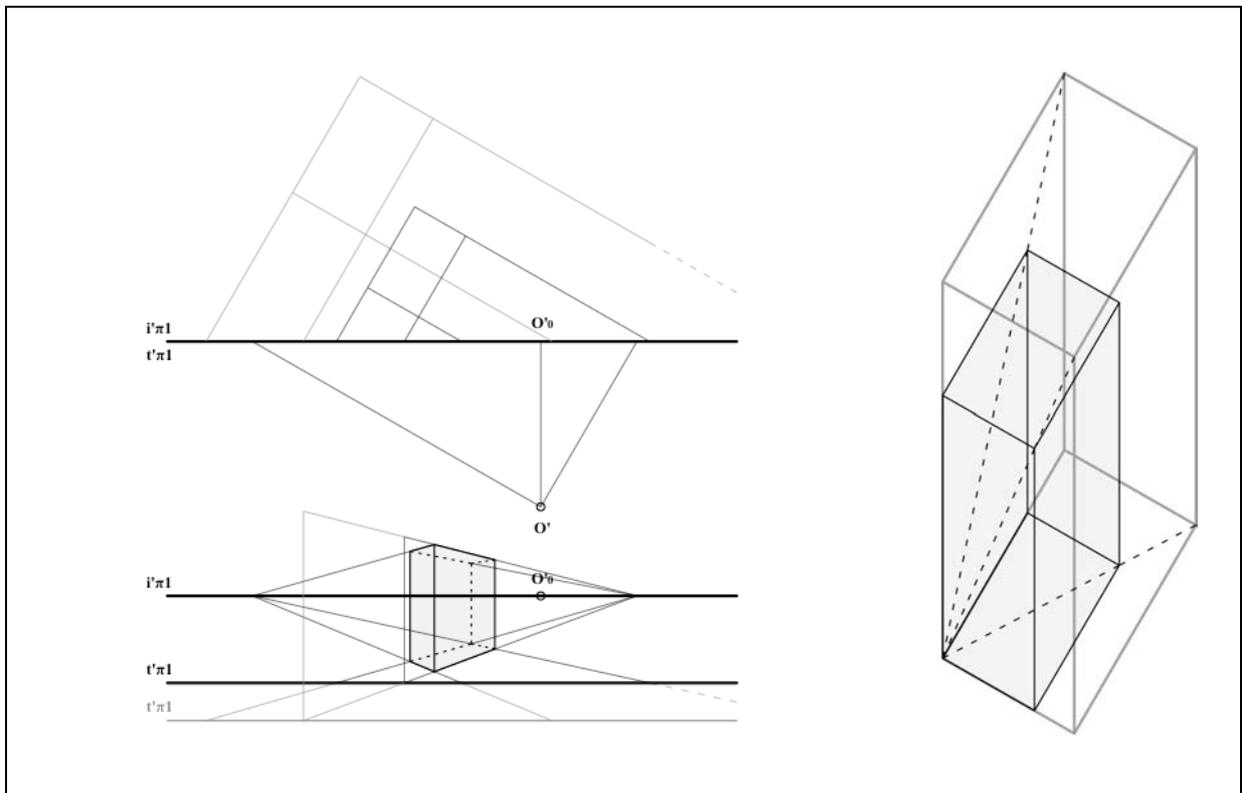
Dato il rettangolo rappresentato in prospettiva nella parte inferiore della figura sottostante, si procede individuando sull'orizzonte le fuga  $I'a$  dei lati orizzontali del rettangolo e la fuga  $I'b$  del segmento ortogonale; condotta da  $I'a$  una retta ortogonale all'orizzonte, si interseca questa con la retta che a cui appartiene la diagonale del rettangolo nella prospettiva, determinando il punto  $P$ ; conducendo da  $P$  una retta che formi col segmento  $I'a-P$  un angolo pari a quello noto  $\vartheta$ , sull'orizzonte si individua per intersezione il punto  $F$ , il cui ribaltamento con centro in  $I'a$  sul semicerchio di diametro  $I'a-I'b$  determina  $O'^*$  e la sua proiezione ortogonale  $O'_0$  sull'orizzonte.



## DETERMINAZIONE DELL'ORIENTAMENTO ESTERNO

Sul quadro della rappresentazione prospettica, la retta fondamentale è la traccia del piano geometricale, di cui la fuga è rappresentata dalla retta d'orizzonte. Il posizionamento della fondamentale rispetto all'orizzonte consente, nei procedimenti manuali, la restituzione delle geometrie in proiezione di Monge secondo il fattore di scala desiderato. Nell'ambito di un processo di restituzione assistito al computer, come mostrato in figura, la posizione della retta fondamentale può essere arbitrariamente assunta –utilizzando una fondamentale fittizia<sup>37</sup>–, in quanto, al variare di questa, non variano i rapporti proporzionali della restituzione, il cui fattore di scala può essere quindi stabilito a posteriori qualora sia nota almeno una dimensione reale. Il *software Brunelleschi* è programmato per assegnare automaticamente la posizione della fondamentale in coincidenza col margine inferiore dell'immagine prospettica.

<sup>37</sup> Pag. 24 di questo stesso testo, pagg. 50-52 da Anna Sgrosso, *Note di Fotogrammetria Applicata all'Architettura*, Napoli, 1979



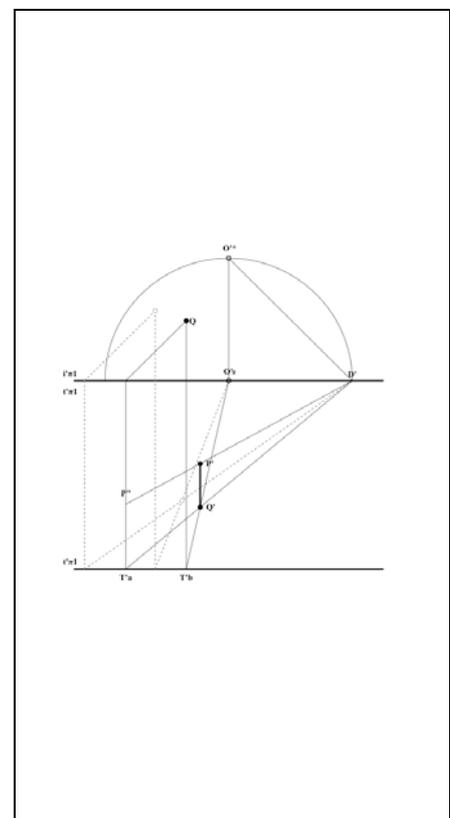
## METODO DI RESTITUZIONE

Noti gli imprescindibili elementi di orientamento della prospettiva, è possibile, attraverso dei procedimenti grafici, trasformare, dalla proiezione centrale a quella ortogonale, per omologia, ciascun punto di cui sia individuabile la proiezione sul geometrale prospettico o che su di esso giaccia.

Il procedimento prescelto e codificato nel *software Brunelleschi*, come mostrato in figura, utilizza il punto principale  $O'_0$  e un punto di distanza  $D'$ , sia  $O'_0-D'$  la distanza principale predeterminata; se si considera il piano orizzontale geometrale, che nella rappresentazione prospettica è delimitato dalla fondamentale  $t'\pi_1$  e dalla retta di orizzonte  $i'\pi_1$ , di ogni punto  $Q'$  su di esso giacente è possibile individuare sulla fondamentale le tracce  $T'b$  e  $T'a$  proiettandolo rispettivamente da  $O'_0$  e  $D'$ .

Condotte due rette, la prima, da  $T'b$ , ortogonale rispetto alla fondamentale, e la seconda, da  $T'a$ , inclinata di  $45^\circ$  ( $\pi/4^{\text{rad}}$ ) rispetto alla stessa, la loro intersezione determina il "punto restituito"  $Q$ , in proiezione ortogonale dall'alto (in figura, per evitare sovrapposizioni,  $T'b$  e  $T'a$  sono traslate dalla  $t'\pi_1$  alla  $i'\pi_1$ ); l'altezza reale del segmento  $P'-Q'$  si può ottenere proiettando  $Q'$  da  $D'$  (o da  $O'_0$ ) e intersecando  $P''$  sulla verticale condotta da  $T'a$  (o da  $T'b$ )<sup>38</sup>.

<sup>38</sup> In figura viene mostrato come, in tempo reale, l'intera procedura rappresenti una sorta di "pantografo" della restituzione prospettica il quale, al variare di  $Q'$  -con un semplice spostamento del mouse- determina la posizione di  $Q$ .

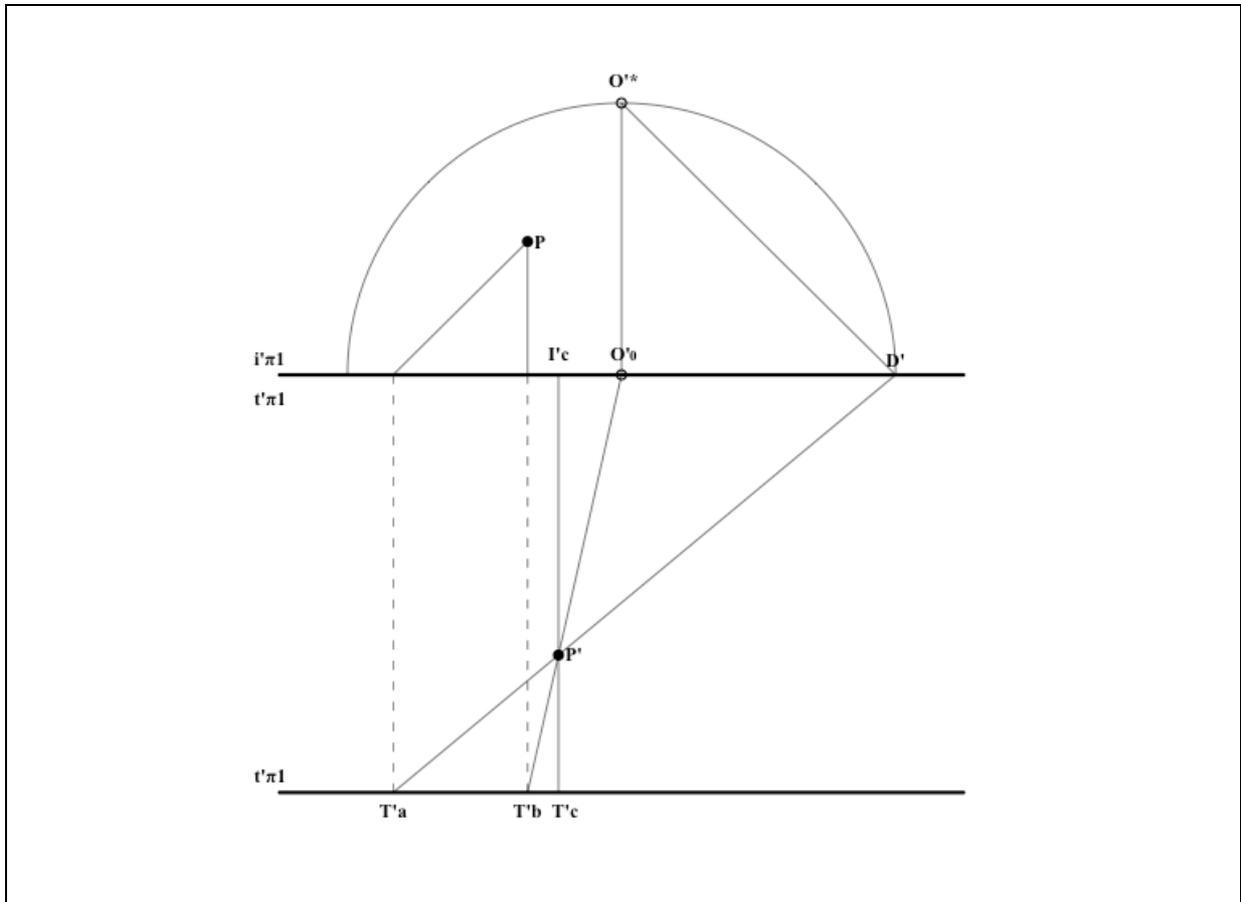


*Restituzione di un punto giacente sul piano  
geometricale<sup>39</sup>*

<sup>39</sup> Scelto il punto  $P'$  da restituire, vengono individuati i punti  $T'c$  e  $I'c$  sulle rette fondamentale e d'orizzonte ( $T'c_x = I'c_x = P'_x$ ); essendo evidente la similitudine fra i triangoli  $T'a-P'-T'c$  e  $D'-P'-I'c$ , ne segue la proporzione  $T'a-T'c : D'-I'c = P-T'c : P-I'c$ , da cui si ricava  $T'a-T'c$  essendo noti gli altri tre valori, e quindi l'ascissa di  $T'a$  rispetto a quella del punto  $P$  ( $T'c$ ); data inoltre la similitudine dei triangoli  $T'b-P-T'c$  e  $O'_0-P-I'c$ , parimenti ne consegue la proporzione  $T'b-T'c : O'_0-I'c = P-T'c : P-I'c$ , da cui si ricava  $T'b-T'c$  essendo noti gli altri tre valori, e quindi l'ascissa di  $T'b$  rispetto a quella del punto  $P$  ( $T'c$ ). La distanza del "punto restituito"  $P'r^*$  dalla retta fondamentale è data dalla seguente espressione  $Pr_y = T'a_x - T'b_x$  (valore assoluto).

Per restituire un punto  $P'$  giacente sul piano geometricale, mi sono servito delle proiezioni ortogonali dello stesso sulle rette fondamentale e di orizzonte, rispettivamente  $T'c$  e  $I'c$ .

La determinazione delle tracce  $T'a$  e  $T'b$  è immediata se si considera che i triangoli  $T'a-P'-T'c$  e  $T'b-P'-T'c$  sono simili ai triangoli noti, rispettivamente,  $D'-P'-I'c$  e  $O'_0-P'-I'c$ ; la distanza del punto restituito  $P$  dalla retta fondamentale è uguale alla distanza che intercorre fra le due tracce  $T'a$  e  $T'b$ .



## CENNI DI PROGRAMMAZIONE

*Brunelleschi* è la “traduzione” -attraverso l’utilizzo di un linguaggio di sviluppo (il *RealBasic*) per personal computer- di alcuni procedimenti grafici di restituzione prospettica in un *software* che ne consenta una più semplice, veloce, corretta applicazione. E’ doveroso in questo testo, dopo aver trattato dei procedimenti di restituzione in vera forma di immagini di oggetti deformate prospetticamente, citare seppur sommariamente alcune nozioni fondamentali sulla programmazione e sui linguaggi ad essa rivolti.

I linguaggi di programmazione sono gli strumenti che consentono, allo sviluppatore di *software*, di “comunicare” con la “macchina” (il *computer*) e far sì che questa svolga un determinato compito. In realtà l’unico linguaggio “recepito” dalla “macchina”, dal suo microprocessore come da qualsiasi altro componente, è il “linguaggio macchina”, cioè una serie non casuale di segnali “acceso-spento” –un “codice binario” consistente in una sequenza di 0 e di 1, se visto dalla parte del programmatore (ciascuno di questi 0 o 1 rappresenta il famigerato *bit*). Ovviamente, nel corso degli anni, si è tentato con alterne fortune di introdurre dei linguaggi “interpretati”, ovvero più intuitivi e comprensibili per il programmatore, che solo alla fine dello sviluppo “compilava”, attraverso un “interprete”, il linguaggio utilizzato in linguaggio macchina. I linguaggi di programmazione, guardati alla luce di quanto detto, vengono suddivisi in linguaggi di “basso livello” e linguaggi di “alto livello”: se si considera un diagramma lineare in cui in “alto” sia il programmatore e in “basso” il *computer*, per linguaggio di basso livello si intende un linguaggio più comprensibile (vicino) alla macchina piuttosto che al programmatore; viceversa, per alto livello ci si riferisce ad uno più intuitivo e comprensibile per il programmatore (il *RealBasic* ne è uno degli esempi).

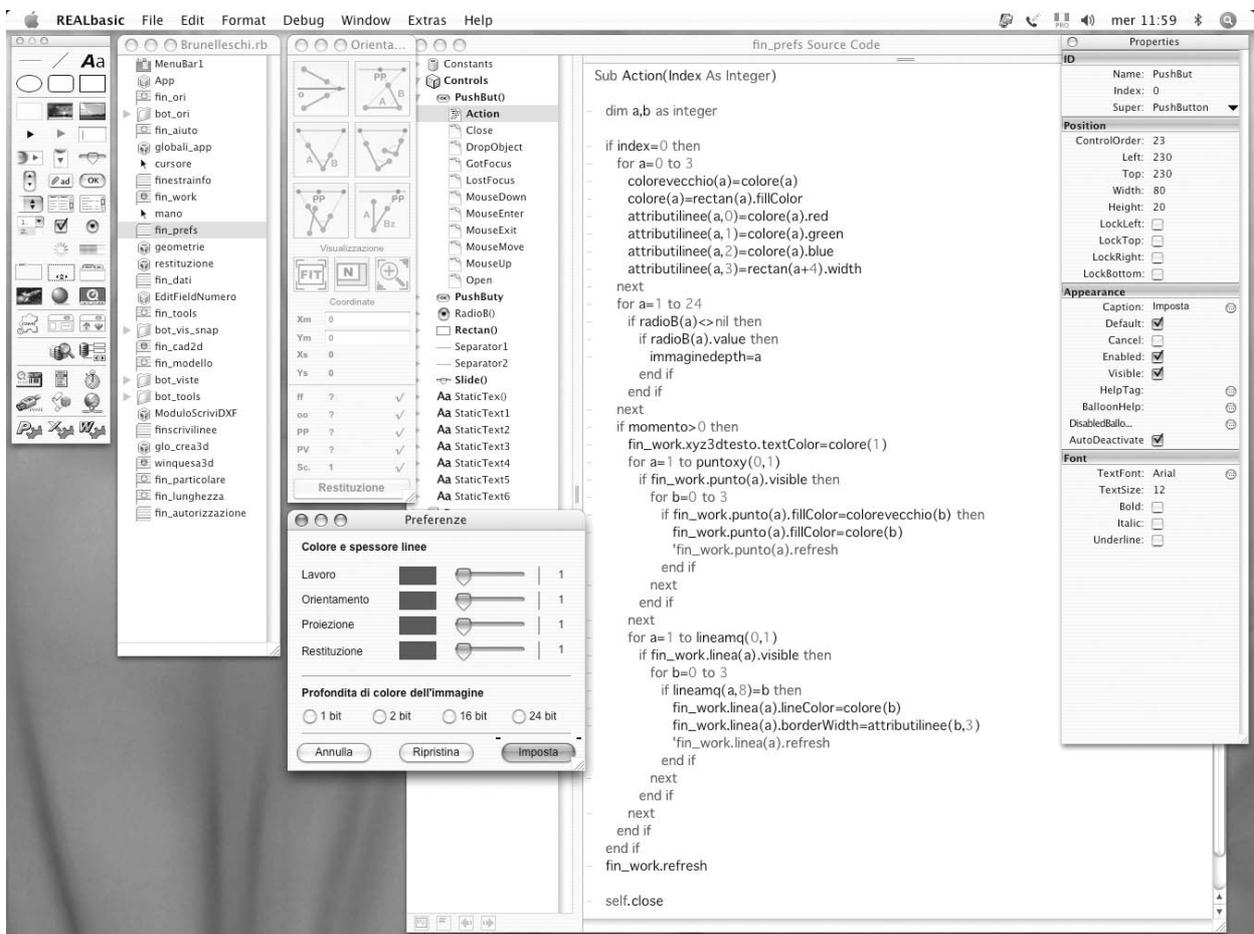
Il linguaggio di basso livello per eccellenza è il linguaggio *Assembly*, il quale altro non è che la rappresentazione simbolica delle cifre binarie che costituiscono il linguaggio macchina. Ogni famiglia di microprocessori (CPU) ha il proprio linguaggio *Assembly* e di conseguenza il proprio linguaggio macchina (codice binario), pertanto un *software* scritto per una specifica macchina, nella quasi totalità dei casi, non funzionerà su altre macchine con diverso microprocessore. Per quanto riguarda invece i linguaggi di alto livello, è possibile dallo stesso codice “compilare” un *software* che funzioni su macchine diverse, se è disponibile un compilatore –il quale altro non è che un *software* a sua volta- che traduca quel codice per diverse piattaforme *hardware* (come nel caso del *RealBasic*).

Il *Fortran*, che deve il suo nome alle iniziali di “formula translation”, fu uno dei primi linguaggi di programmazione di alto livello. Ideato intorno al 1950 da John Backus, un dipendente della IBM, il *Fortran* era rivolto allo sviluppo di calcoli matematici e scientifici, il suo utilizzo ebbe una grande diffusione e sulla sua scia vennero progettati moltissimi altri linguaggi. Molti studiosi, tra i quali lo stesso Backus, si interessarono allo sviluppo dell'*Algol*, che non ebbe però lo stesso successo ottenuto dal *Fortran*. Maggior fortuna ebbe invece il *Cobol* (*common business oriented language*, ossia linguaggio rivolto alla gestione degli affari), nato nel 1960 principalmente per sviluppare programmi gestionali, cioè atti alla risoluzione di problemi aziendali; la sua enorme diffusione fu favorita dalla sua discreta facilità, dovuta anche al fatto che le istruzioni somigliano molto a frasi inglesi, trovando una grandissima applicazione nelle aziende. Negli anni successivi nacque il *Basic* (*beginner's all purpose symbolic instruction code*, cioè codice generico di istruzioni simboliche per principianti), grazie al progetto di Kurtz e Kemeny; a tutt'oggi è tra i migliori linguaggi utilizzati nelle scuole a scopo didattico e, dato il favore

incontrato, ne sono state realizzate molte versioni per apportare miglioramenti: dal *Basic* discende tutto il filone dei vari *Basic* di tipo "visuale", "orientati agli oggetti" e con l'interfaccia grafica (bottoni, finestre, etc.) di immediato utilizzo per lo sviluppatore (la più recente versione di *Basic* visuale è il *RealBasic* della *Real Software*). Niklaus Wirth ideò e realizzò il *Pascal* affinché facilitasse l'applicazione delle regole e delle tecniche di programmazione. Il suo scopo era ottenere un linguaggio adatto per l'insegnamento della scrittura di programmi e centrò benissimo il suo obiettivo, tanto che il *Pascal* è ancora oggi molto usato nelle scuole. Nel 1972 Dennis Ritchie realizzò la prima versione del linguaggio *C* che si distingueva dai suoi predecessori per il fatto di implementare una vasta gamma di tipi di dati come carattere, interi, numeri in virgola mobile, strutture etc. Da allora il linguaggio non ha subito profonde trasformazioni: la sua sintassi è stata estesa, soprattutto in conseguenza della programmazione orientata agli oggetti (*C++*), ma nella sostanza il linguaggio è rimasto quello delle origini. Il *C* è un linguaggio ad alto livello che possiede un insieme ristretto di costrutti di controllo e di parole chiave, e un ricco insieme di operatori, ed è un linguaggio apparentemente povero: è stato talvolta definito come "il linguaggio di più basso livello tra i linguaggi ad alto livello", infatti nasce per lo sviluppo di sistemi operativi, quindi per *software* di basso livello, ma riesce a mantenere una certa semplicità d'uso. Il trucco usato dai suoi sviluppatori per mantenere compatto il linguaggio, e nel contempo estenderne le funzionalità, sta nell'affidare le funzioni più complesse a un'insieme di librerie esterne. All'interno del mondo informatico, il successo del linguaggio *C* è universalmente riconosciuto. Il suo trampolino di lancio è stato il sistema operativo *Unix*, scritto proprio con questo linguaggio (*Unix* è stato il primo sistema operativo moderno *Open Source*, cioè a codice aperto).

Java, la cui creazione risale a metà degli anni '90, è stata un'invenzione rivoluzionaria ed indispensabile nel mondo della programmazione, non solo per i *browser* (*software* che consentono, fra l'altro, la navigazione su *internet*). La SUN, qualche anno fa presentò il Java come un linguaggio di programmazione object oriented semplice e familiare a chi conosce il C++, indipendente dall'architettura, e sicuro per l'uso in rete. Essendo un linguaggio multiplatforma, il compilatore non crea degli applicativi eseguibili dal computer, ma dei file che devono essere interpretati dalla JVM (*Java virtual machine*, un microprocessore virtuale) implementata sui vari sistemi -ciò ha sempre reso l'esecuzione di un'applicazione Java molto più lenta dei programmi normali.

40 L'ambiente di sviluppo *RealBasic* su piattaforma Apple, sistema operativo Mac OS X 10.4.7

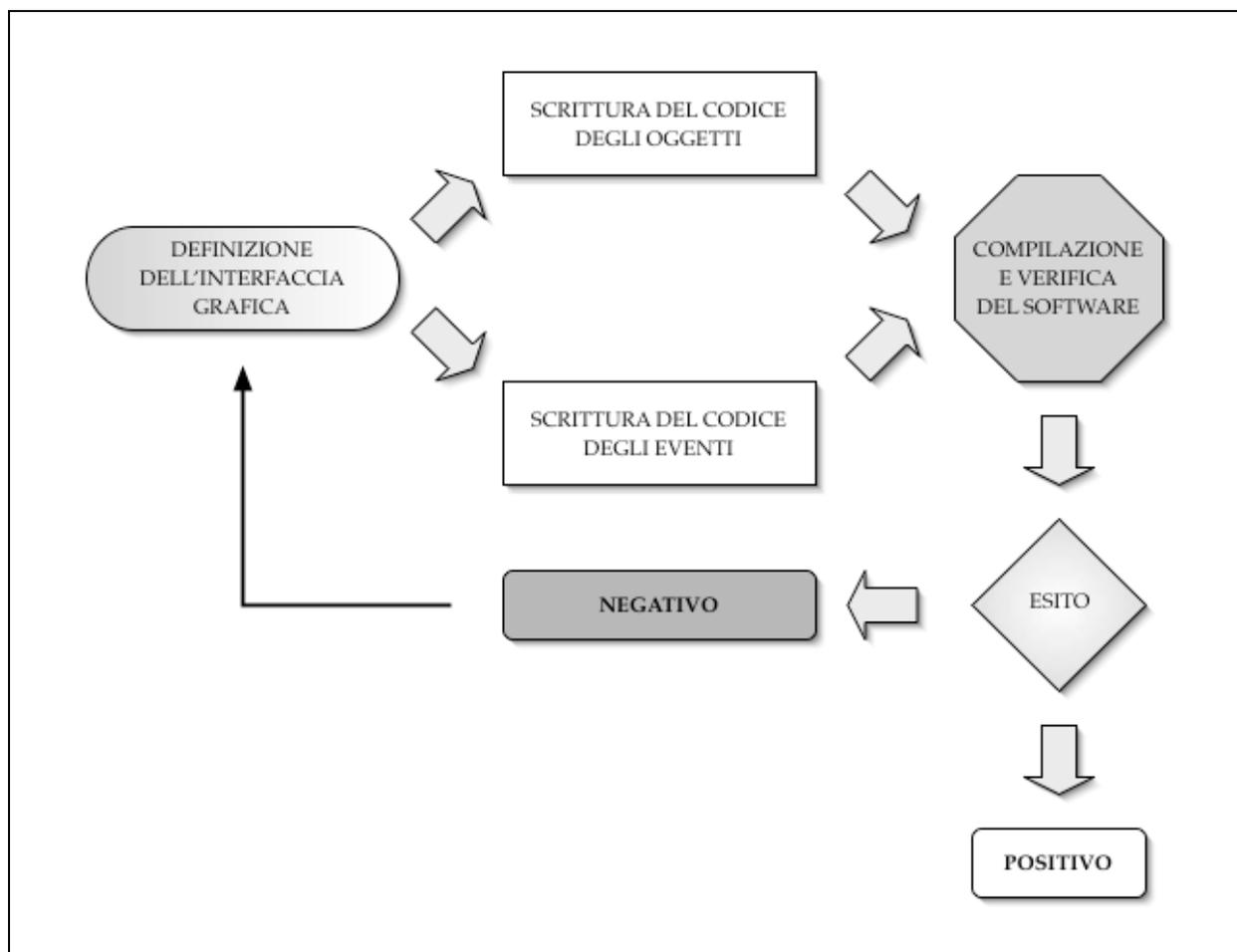


## *Programmazione con RealBasic*

L'ambiente di sviluppo<sup>40</sup> utilizzato per lo sviluppo del *software Brunelleschi* è stato il *RealBasic*, essendo esso la più recente e moderna versione del noto linguaggio *Basic* che lo scrivente utilizza dal 1980. Il *RealBasic*, nella sua attuale versione disponibile, consente lo sviluppo e la produzione di *software* multiplatforma, cioè in grado di essere utilizzati su sistemi operativi e configurazioni di personal computer differenti -Microsoft Windows, Apple MacOS X, Linux.

*RealBasic*, come tutti i moderni linguaggi di programmazione, è un linguaggio *object-oriented*, "orientato agli oggetti", cioè un linguaggio in cui è semplificato l'utilizzo degli "oggetti", i quali altro non sono che gli elementi grafici di interfaccia dei recenti sistemi operativi come di tutti o quasi i *software* utilizzabili: i pulsanti, i menù a tendina, le freccette di scorrimento, le finestre; se lo sviluppatore ha necessità di creare un comando del tipo "bottone", il quale una volta premuto dall'utente finale emetta un suono, può con molta semplicità utilizzare uno degli oggetti "bottone" già predisposti dall'ambiente di sviluppo (*RealBasic* nel nostro caso) e semplicemente scrivere il codice -in grado di generare il suono voluto- nell'*evento* "bottone premuto" del relativo oggetto. L'accenno al termine "evento" denuncia che in realtà *RealBasic* è anche un linguaggio "orientato agli eventi" (*event-driven*), cioè consente di scrivere con immediatezza il codice in una serie di "eventi" già previsti in fase di sviluppo; "eventi" sono ciascun avvenimento che può intercorrere durante l'utilizzo di un software, sia per intervento dell'utente finale, sia in maniera indipendente; sono "eventi" il trascorrere di una certa quantità di secondi, il movimento del mouse da parte dell'utente finale, la chiusura di una certa finestra, la pressione di un bottone di comando.

Il processo di sviluppo del *software Brunelleschi* si può sintetizzare come percorso, non lineare e recursivo, di: definizione dell'interfaccia grafica del *software* e degli "oggetti" che la compongono; selezione degli "eventi" che potranno caratterizzare l'uso del *software* –fra cui quelli "causati" dall'utente finale; scrittura del codice di ciascun "evento", cosicchè ciascun evento produca la corretta "risposta" del *software* –fra le altre cose, trascrizione in codice *RealBasic* delle formule di geometria analitica in precedenza trattate; compilazione del *software* come applicativo autonomo e verifica del suo corretto funzionamento.



***BRUNELLESCHI MANUALE OPERATIVO***



## LICENZA D'USO DEL SOFTWARE BRUNELLESCHI

Il *software Brunelleschi*, Tesi di Dottorato di Ricerca in Teoria e Storia della Rappresentazione, XIX ciclo, è stato realizzato dallo scrivente arch. Francesco Vinci che ne detiene i diritti d'autore e la proprietà esclusiva.

È consentito l'uso libero e gratuito del *software* per effettuare studi di restituzione prospettica secondo quanto previsto dal programmatore –lo scrivente.

Ciascuno studio di restituzione prospettica potrà essere liberamente pubblicato a condizione di citare l'utilizzo del *software Brunelleschi*, includendo le seguenti informazioni: **«sviluppato dall'arch. Francesco Vinci per il Dipartimento ASTRA della Facoltà di Architettura di Siracusa, direttore prof. arch. Giuseppe Pagnano - Università degli Studi di Catania»**; l'uso del *software* è inoltre sottoposto alla richiesta e all'ottenimento dei codici di autorizzazione, forniti gratuitamente, individuali per ciascun *file* di immagine su cui l'utilizzatore intende effettuare la restituzione prospettica assistita.

La riproduzione del *software*, la sua diffusione, traduzione, copia, memorizzazione su supporto magnetico e la trasmissione per via elettronica, per fini non commerciali, sono permessi solo se accompagnati dalla presente licenza d'uso del *software*.

Ciascun altro uso del *software* e della documentazione ad esso associata non è consentito, in particolar modo modificare, tradurre, riprogettare, decompilare, deassemblare il *software* e la documentazione ad esso associata, senza autorizzazione scritta dello scrivente.

Il codice sorgente del *software* è di proprietà esclusiva dello scrivente.

La presente licenza ha validità per tutto il periodo di utilizzo del *software* e si estingue con la distruzione del *software* e della documentazione ad esso associata così come di tutte le eventuali copie prodotte .

Sebbene lo sviluppatore abbia preso tutti gli accorgimenti possibili, non garantisce che il software, o qualsiasi informazione in esso contenuta o visualizzata, saranno privi di errori o incontreranno le esigenze dell'utilizzatore, oppure avranno un funzionamento illimitato e privo di errori o che gli eventuali errori presenti nel *software* potranno essere corretti.

Il software viene installato dall'utente a proprio rischio e in nessun caso lo sviluppatore potrà essere ritenuto responsabile di qualsiasi perdita o danno, compreso la perdita di profitto o altre perdite risultanti dalla incapacità di utilizzo del *software*, o per mancanze o errori del *software* stesso, oppure dall'uso del *software* in associazione con qualsiasi programma.

È responsabilità dell'utilizzatore effettuare una verifica sul *software* per evidenziare la eventuale presenza di *virus* informatici, sebbene lo sviluppatore abbia preso le necessarie precauzioni per evitare la presenza di *virus*.

Lo scrivente non si assume alcuna responsabilità per perdite, danni o costi di qualsiasi genere, causati dalla presenza di *virus*.

E' consentito installare il *software* e utilizzarlo su un *computer* di vostra proprietà, presso la vostra sede. E' inoltre possibile trasferire il *software* da un *computer* ad un altro, usare il *software* su un qualsiasi *computer* o sistema che consenta a più utenti di accedere al *software* stesso, usare il *software* esclusivamente secondo i termini della presente licenza d'uso.

L'installazione del *software* comporta l'accettazione dei termini della presente licenza.

## REQUISITI DI SISTEMA

Lo sviluppo di *Brunelleschi* è stato integralmente condotto e portato a termine dallo scrivente nel proprio triennio di Dottorato di Ricerca in “Teoria e Storia della Rappresentazione”. Il *development* è stato eseguito con l’ambiente di sviluppo *RealBasic* (v.5.5) su piattaforma Apple PowerMac G4 (*hardware dual cpu 1.4ghz*, sistema operativo Mac OS X 10.4.7), così come la breve, –per motivi contingenti- seppur accurata fase di *testing* del *software*. *Brunelleschi* nella sua versione 1.00 è pertanto da considerarsi un applicativo stabile ma in fase “beta” di sviluppo, e i requisiti minimi di sistema riportati di seguito puramente indicativi, data l’impossibilità di verificare il corretto funzionamento del software su molteplici configurazioni di *computer-system*.

### *Apple Mac OS X*

Qualsiasi Apple Macintosh con processore PowerPC G3, G4, G5 (compatibilità CPU Intel non verificata), spazio libero su disco 500 Mb, memoria RAM libera 128 Mb, risoluzione monitor 1024x768 punti o superiore, profondità di colore 24 bit (milioni di colori).

Sistema Mac OS X 10.2 o superiore, Apple QuickTime 5 o superiore, Quesa Lib (inclusa nell’archivio).

### *Microsoft Windows*<sup>41</sup>

Qualsiasi hardware con processore Intel o AMD a 500mhz o superiore, spazio libero su disco 500 Mb, memoria RAM libera 128 Mb, risoluzione monitor 1024x768 punti o superiore, profondità di colore 24 bit (milioni di colori).

Sistema Microsoft Windows 98/ME/2000/XP/2003, Apple QuickTime 5 o superiore, Quesa Lib (inclusa nell’archivio).

<sup>41</sup> La versione per Microsoft Windows può presentare malfunzionamenti o *bug* non presenti nella *release* per Apple Mac OS X; se possibile, utilizzare *Brunelleschi* per Mac OS X su piattaforma Apple.

## INSTALLAZIONE, DISINSTALLAZIONE E AVVIO

*Brunelleschi* viene fornito senza *installer* e, se il sistema ospite soddisfa i requisiti minimi richiesti, può essere eseguito in "prova" anche direttamente dal *cd-rom*. La normale procedura di utilizzo di *Brunelleschi* prevede la semplice copia dalla cartella relativa (dentro "Macintosh" o "Windows") contenuta dal *cd-rom Brunelleschi* al disco fisso di destinazione. *Brunelleschi* non copia alcun *file* nel disco rigido dell'utente al di fuori della propria cartella/*directory*, e la rimozione della stessa consente la completa "disinstallazione" del *software* dal proprio *computer*. Per l'utilizzo del *software*, avviare l'applicazione *Brunelleschi* (.app o .exe) contenuta all'interno della cartella/*directory* di installazione. Per il funzionamento del motore di visualizzazione tridimensionale integrato, può essere necessario procedere a quanto di seguito indicato.

### *Apple Mac OS X*

Quesa Lib installazione: copiare il file "**Quesa**" dalla cartella di installazione di *Brunelleschi* dentro **/Libreria/CFMSupport/** (presente nel primo livello del disco con installato Mac OS X)

### *Microsoft Windows*

Quesa Lib installazione: copiare il file "**Quesa.dll**" dalla cartella di installazione di *Brunelleschi* dentro **windows/system** (cartella "system" situata nella cartella "windows"; se il file "Quesa.dll" non è visibile, attivare l'opzione di windows di visualizzazione di tutti i file; se l'opzione non è attiva, tutti i file di sistema sono invisibili).

## INFORMAZIONI PRELIMINARI

La restituzione prospettica assistita con l'utilizzo del *software Brunelleschi* può essere sintetizzata in quattro distinte fasi di lavoro -le prime due propedeutiche alle altre- di seguito elencate:

### ***Importazione della rappresentazione prospettica,***

consistente nelle operazioni atte ad ottenere l'immagine prospettica di cui si vuole eseguire la restituzione in formato digitale di tipo *raster*;

### ***Determinazione dell'orientamento della prospettiva,***

consistente nell'individuazione, con un'operazione di "ricalco" sulla rappresentazione prospettica, degli elementi geometrici tali da consentire la determinazione dell'orizzonte, del punto di vista, del punto principale e della distanza principale, e con essi lo "spazio tridimensionale" della prospettiva;

### ***Definizione degli elementi geometrici della prospettiva,***

consistente nell'individuazione, con un'operazione di "ricalco" sulla rappresentazione prospettica, delle forme geometriche di sintesi riconosciute dall'utente, rette, piani, o volumi, fra quelle messe a disposizione dal *software* per la restituzione automatica; in questa fase è consentito in qualsiasi momento passare alla successiva;

### ***Gestione del modello restituito,***

consistente nella visualizzazione, controllo e verifica, del modello tridimensionale restituito, nelle sue proiezioni ortogonali e assonometriche, nella sua "realtà" spaziale, compresa l'esportazione del modello o degli elaborati bidimensionali verso altri *software* per successive elaborazioni; in questa fase è consentito in qualsiasi momento passare alla fase precedente.

## L'AMBIENTE DI LAVORO

Per chiarificare in maniera efficace l'utilizzo del *software Brunelleschi* è opportuno in primo luogo descrivere quello che è comunemente definito "ambiente di lavoro" dell'applicativo<sup>42</sup>.

Lo spazio operativo a disposizione dell'utente sarà così suddiviso (vedi figura in basso):

1) barra dei menu a tendina<sup>43</sup>, con i comandi principali di *Brunelleschi*;

2) finestra di informazioni e aiuto, fornisce suggerimenti in tempo reale e consente di inviare comandi immediati o altri dati richiesti dal *software*;

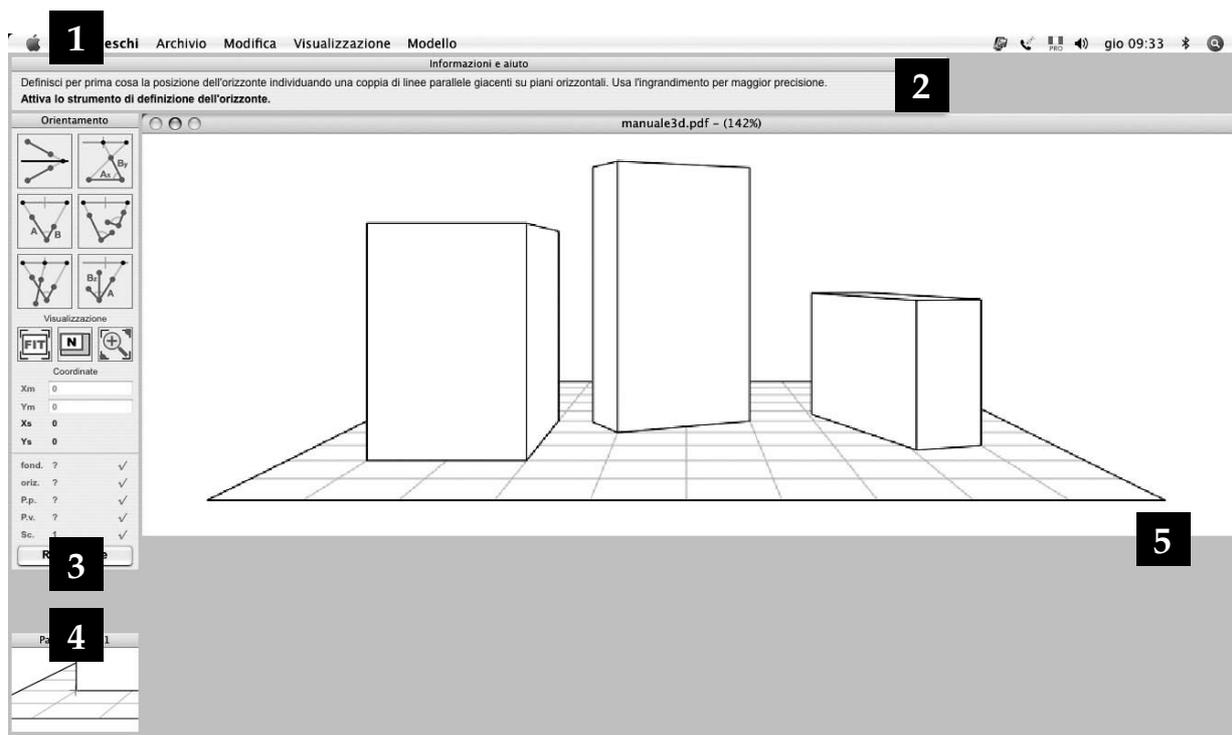
3) finestra degli strumenti principali, cangiante, a seconda della fase di lavoro, in orientamento, restituzione e modello;

4) finestra di visualizzazione del particolare, in rapporto 1:1 fra *image pixels/monitor pixels*;

5) finestra del "foglio" di lavoro, su cui l'utente opererà per procedere alle fasi della restituzione;

<sup>42</sup> L'ambiente di lavoro e le successive descrizioni d'utilizzo si riferiscono alla versione di *Brunelleschi* per Apple Mac OS X, identica a quella per Microsoft Windows a meno di alcune minime differenze d'interfaccia grafica; tali differenze, qualora presenti, saranno dichiarate in nota.

<sup>43</sup> Su piattaforma Microsoft Windows la barra dei menu è contenuta nella parte alta della finestra (2) di informazioni e aiuto



## IMPORTAZIONE DELLA RAPPRESENTAZIONE PROSPETTICA

La restituzione assistita con *Brunelleschi* si basa sulla disponibilità di un'immagine prospettica a quadro verticale, in formato digitale di tipo *raster* –prodotta al computer, o acquisita tramite *scanner* o fotocamera digitale. Attraverso Apple QuickTime 5 o superiore (requisito minimo di sistema) è possibile importare un file di immagine di tipo BMP, PICT, PhotoShop, JPEG, PNG, SGI, TGA, TIFF o QuickTime image.

A discrezione dell'utente, l'immagine può essere corredata –prima della sua importazione in *Brunelleschi*- di elementi di riferimento quali assi o linee di costruzione o punti notevoli, non presenti in prima istanza nella prospettiva, attraverso uno dei vari applicativi di *editing* delle immagini fra i *software* disponibili sul mercato. Questa "preparazione" dell'immagine, seppur non necessaria, può spesso rendere più agevole la fase di restituzione con l'uso del *software Brunelleschi*.

Per importare un'immagine nel foglio di lavoro di *Brunelleschi*, è possibile selezionare il comando *Nuova restituzione* dal menu *Archivio*, o premere *command+N*<sup>44</sup> sulla tastiera, o cliccare il "collegamento" (*link*) disponibile dalla finestra di informazioni –la quale fornisce una guida al *software* in tempo reale. La finestra di selezione file che apparirà a seguito di tale operazione, consentirà di sfogliare il contenuto dei supporti magnetici o ottici (*hard disk, cd-rom, o altro*) su cui è memorizzata l'immagine prospettica di tipo *raster* -secondo uno dei formati riconoscibili dal *software Brunelleschi* attraverso Apple QuickTime- e di importarla selezionandola e premendo il bottone di comando *Apri*.

Per poter procedere con la restituzione, è necessario ottenere il codice di autorizzazione, collegato al codice di richiesta fornito da *Brunelleschi* all'atto di importazione dell'immagine, univoco per ciascun *file* di immagine.

**Per importare un' immagine in formato digitale selezionare il comando "Nuova restituzione" dal menu "Archivio"**

<sup>44</sup> CTRL+N su piattaforma Microsoft Windows

Quando il *software* mostrerà una finestra analoga a quella in figura in fondo alla pagina, prendere nota del **codice di richiesta**.

**Annotare il codice di richiesta e richiedere via e-mail, o posta, il codice di autorizzazione**

L'uso del *software Brunelleschi* è gratuito, e ciascuno studio di restituzione prospettica con esso eseguito potrà essere liberamente pubblicato a condizione di citarne l'utilizzo, includendo le seguenti informazioni: «sviluppato dall'arch. Francesco Vinci per il Dipartimento ASTRA della Facoltà di Architettura di Siracusa, direttore prof. arch. Giuseppe Pagnano - Università degli Studi di Catania».

Per ottenere gratuitamente il codice di autorizzazione (individuale per ciascun *file* di immagine) è sufficiente farne motivata richiesta tramite *e-mail*<sup>45</sup> all'indirizzo [brunelleschi@arcobaleno2001.it](mailto:brunelleschi@arcobaleno2001.it), specificando i dati personali (nome, cognome, luogo e data di nascita), illustrando brevemente lo studio da svolgere, e **indicando esattamente il codice di richiesta**.

<sup>45</sup> E' possibile richiedere il codice di autorizzazione per posta scrivendo a: arch. Francesco Vinci, Via Sandro Pertini 22, 96012 Avola (SR).

**Autorizzazione**

**Richiesta del codice di autorizzazione**

Ciascuno studio di restituzione prospettica assistito dall'uso del software "Brunelleschi" potrà essere liberamente pubblicato a condizione di citarne l'utilizzo, includendo le seguenti informazioni: sviluppato dall'arch. Francesco Vinci per il Dipartimento ASTRA della Facoltà di Architettura di Siracusa, direttore prof. arch. Giuseppe Pagnano - Università degli Studi di Catania.

Selezionate il collegamento sottostante per richiedere via e-mail il codice di autorizzazione gratuito, specificando i propri dati personali (nome, cognome, luogo e data di nascita), illustrando brevemente lo studio da svolgere, e soprattutto indicando esattamente il codice di richiesta.

**INVIA EMAIL DI RICHIESTA DEL CODICE DI AUTORIZZAZIONE**

Codice di richiesta

Codice di autorizzazione

E' possibile richiedere il codice di autorizzazione per posta scrivendo a:  
Dipartimento ASTRA - Università degli Studi di Catania  
Facoltà di Architettura, via Maestranza 99  
96100 Siracusa

**Esci**

## DETERMINAZIONE DELL'ORIENTAMENTO DELLA PROSPETTIVA

Importata l'immagine prospettica, l'utente si troverà nell'ambiente di lavoro illustrato precedentemente (pag.60). Le principali operazioni che l'utente dovrà, o potrà, eseguire sull'immagine prospettica in questa fase, sono attivabili dalla finestra degli strumenti principali di orientamento (fig.1), alla sinistra del foglio di lavoro.

### *Finestra degli strumenti principali di orientamento.*

La *palette* dei comandi di orientamento, in figura qui a fianco, è suddivisa in tre sezioni; nella sezione in alto sono presenti i sei comandi di determinazione degli elementi di orientamento, che verranno individualmente illustrati successivamente.

Alla sezione denominata "Visualizzazione" appartengono i tre comandi di gestione del fattore di ingrandimento dell'immagine prospettica nel foglio di lavoro; da sinistra a destra: il comando "FIT" mostra l'immagine in modo tale che appaia integralmente sullo schermo; il comando "N" mostra l'immagine (o parte di essa, per immagini di grande risoluzione) in *pixel* reali, cioè in modo tale che il rapporto *image pixels/monitor pixels* sia di 1:1 (questo è il fattore di ingrandimento ideale di lavorare sull'immagine in *Brunelleschi*); il comando "+" consente di ingrandire una porzione dell'immagine indicando su di essa un rettangolo contenente la parte da visualizzare, cliccando due volte per definirne gli estremi della diagonale.

La sezione delle "Coordinate", nella parte alta, consente di controllare le coordinate del puntatore (*mouse*) sull'immagine;  $X_m$  e  $Y_m$  in pixel reali del monitor,  $X_s$  e  $Y_s$  in pixel reali immagine. Più in basso sono mostrate le informazioni sulle coordinate calcolate da *Brunelleschi* per la fondamentale, l'orizzonte, il punto principale, il punto di vista, il fattore di scala.

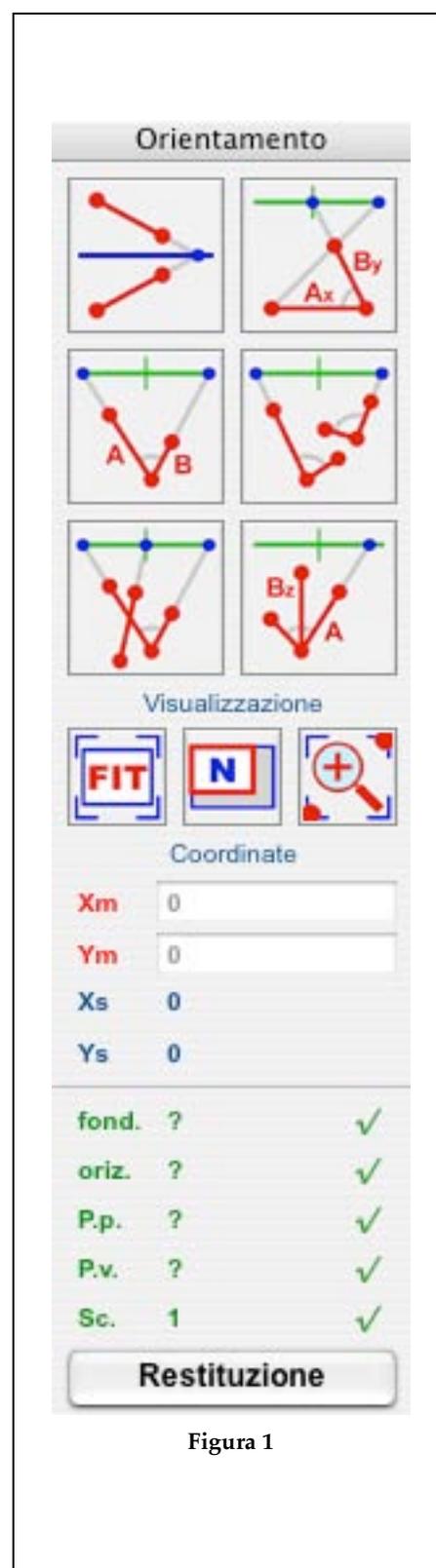
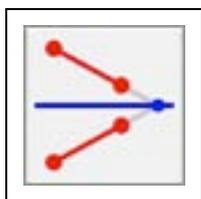


Figura 1



**Determinare l'orizzonte con lo strumento di individuazione di due segmenti fra loro paralleli nella realtà**

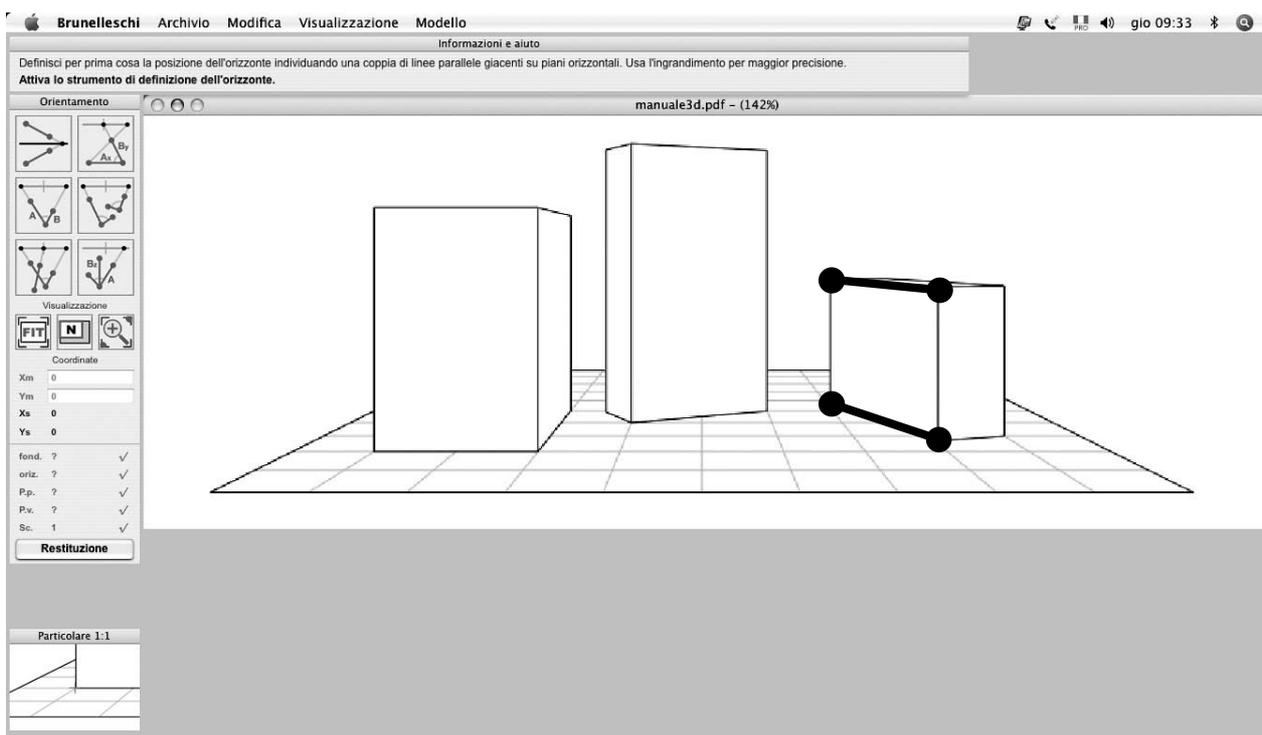
**Individuazione dell'orizzonte<sup>46</sup>.**

La finestra di informazioni segnalerà che la prima operazione da compiere, per determinare l'orientamento della prospettiva, sarà quella di individuare la retta di orizzonte; è possibile attivare questo comando cliccando sul *link*, o sul comando mostrato qui a fianco (in alto a sinistra della finestra degli strumenti, fig.1, pag. prec.).

Attivato il comando, l'utente dovrà indicare sull'immagine prospettica, attraverso quattro *click* del *mouse* (fig. in basso), due segmenti, fra di loro paralleli nella realtà, appartenenti a piani orizzontali; qualora tali segmenti non fossero individuabili, e l'orizzonte fosse presente nell'immagine, è possibile fare "incrociare" due segmenti qualsiasi avendo cura che l'intersezione ricada sull'orizzonte visibile.

Il comando di individuazione dell'orizzonte può essere attivato più volte, utilizzando segmenti differenti, per aumentarne l'accuratezza della coordinata "y" trovata : il *software* farà la media fra le diverse coordinate calcolate. Definito l'orizzonte, gli altri strumenti di determinazione dell'orientamento diventano attivi.

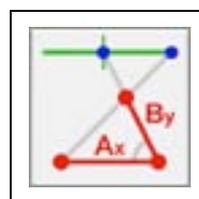
<sup>46</sup> Trattazione dell'argomento in termini geometrici in questo stesso testo, pag.37



*Determinazione del punto di vista e del punto principale, dato un rettangolo di proporzioni note e con due lati paralleli al quadro<sup>47</sup>.*

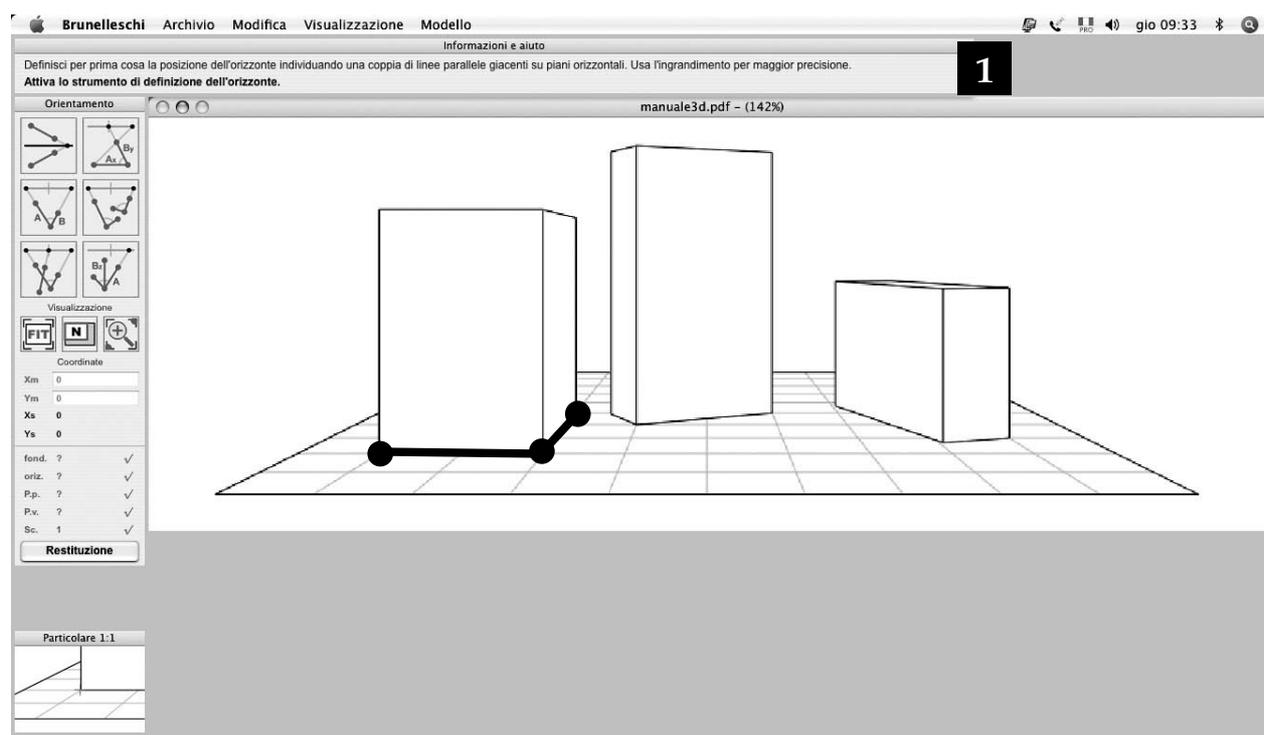
Per la determinazione del punto di vista e del punto principale, il primo dei procedimenti predisposti dal software si basa sull'individuazione sulla prospettiva della deformata di un rettangolo di proporzioni (o dimensioni) note, giacente su un piano orizzontale, con due lati paralleli al quadro.

Attivato, dalla *palette* degli strumenti principali di orientamento (pag.63), il comando il cui simbolo è mostrato qui a fianco, facendo *click* con mouse su di esso, l'utente dovrà indicare sull'immagine, attraverso tre *click* del *mouse*, due lati consecutivi del rettangolo, di cui il primo sia quello parallelo al quadro e il successivo quello ad esso ortogonale (vedi figura in basso). Eseguita questa operazione, nella finestra di informazioni (1), verranno mostrati i campi di testo, indicati come A e B, dove sarà necessario inserire le proporzioni relative (o dimensioni reali) dei due lati; specificare, se desiderato, l'unità di misura; fare click sul bottone "assegna".

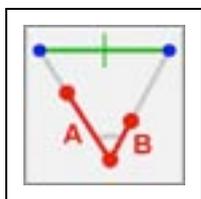


**Determinare il punto di vista e il punto principale utilizzando uno dei comandi disponibili**

<sup>47</sup> Trattazione dell'argomento in termini geometrici in questo stesso testo, pag.38



***Determinazione del punto di vista e del punto principale, dato un rettangolo di proporzioni note con lati non paralleli al quadro<sup>48</sup>.***



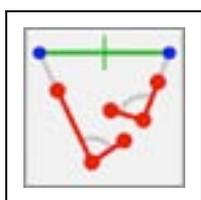
È possibile se sull'immagine prospettica è presente la deformata di un rettangolo di proporzioni (o dimensioni) note, ruotato e giacente su un piano orizzontale.

Attivato il comando (a fianco), l'utente dovrà indicare sull'immagine, attraverso tre *click* del *mouse* (fig. in basso, base di uno qualsiasi dei due parallelepipedi), due lati consecutivi del rettangolo; successivamente, nella finestra di informazioni (1), sarà necessario inserire le proporzioni relative (o dimensioni) dei due lati (vedi pag. prec.).

<sup>48</sup> Trattazione dell'argomento in termini geometrici in questo stesso testo, pag.39

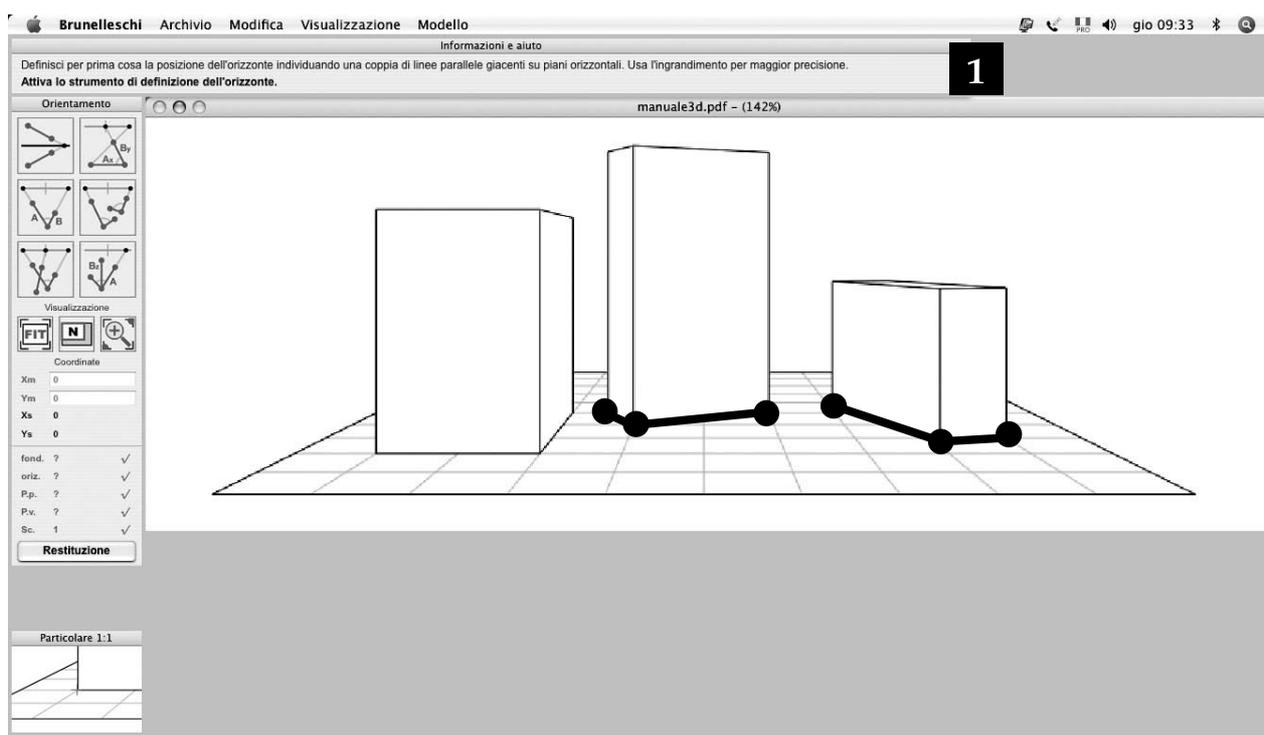
<sup>49</sup> Come sopra, pag.40

***Determinazione del punto di vista e del punto principale, dati due rettangoli diversamente orientati, con lati non paralleli al quadro<sup>49</sup>.***



È possibile se sull'immagine prospettica sono presenti due angoli di 90° nella realtà, diversamente ruotati e con lati incidenti al quadro, giacenti su piani orizzontali.

Attivato il comando (a fianco), l'utente dovrà indicare i due angoli sull'immagine, attraverso sei *click* del *mouse* (fig. in basso).



*Determinazione del punto di vista e del punto principale, dati un rettangolo con lati non paralleli al quadro e un segmento orizzontale ortogonale al quadro<sup>50</sup>.*

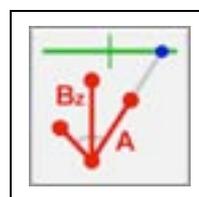
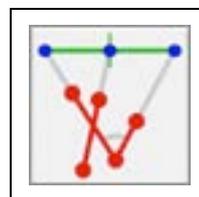
È possibile se sull'immagine prospettica è presente un angolo di  $90^\circ$  (nella realtà), giacente su un piano orizzontale e definito da lati incidenti al quadro, oltre ad un segmento orizzontale che vada in fuga nel punto principale (fig. nella pagina successiva, sulla sinistra).

Attivato il comando il cui simbolo è mostrato qui a fianco, l'utente dovrà indicare l'angolo retto succitato sull'immagine, attraverso tre *click* del *mouse*. Successivamente, con due *click* del *mouse*, dovrà individuare gli estremi del segmento orizzontale ortogonale al quadro.

*Determinazione del punto di vista e del punto principale, dati un rettangolo di proporzioni note, ortogonale al piano orizzontale, e un segmento orizzontale ortogonale ad esso<sup>51</sup>.*

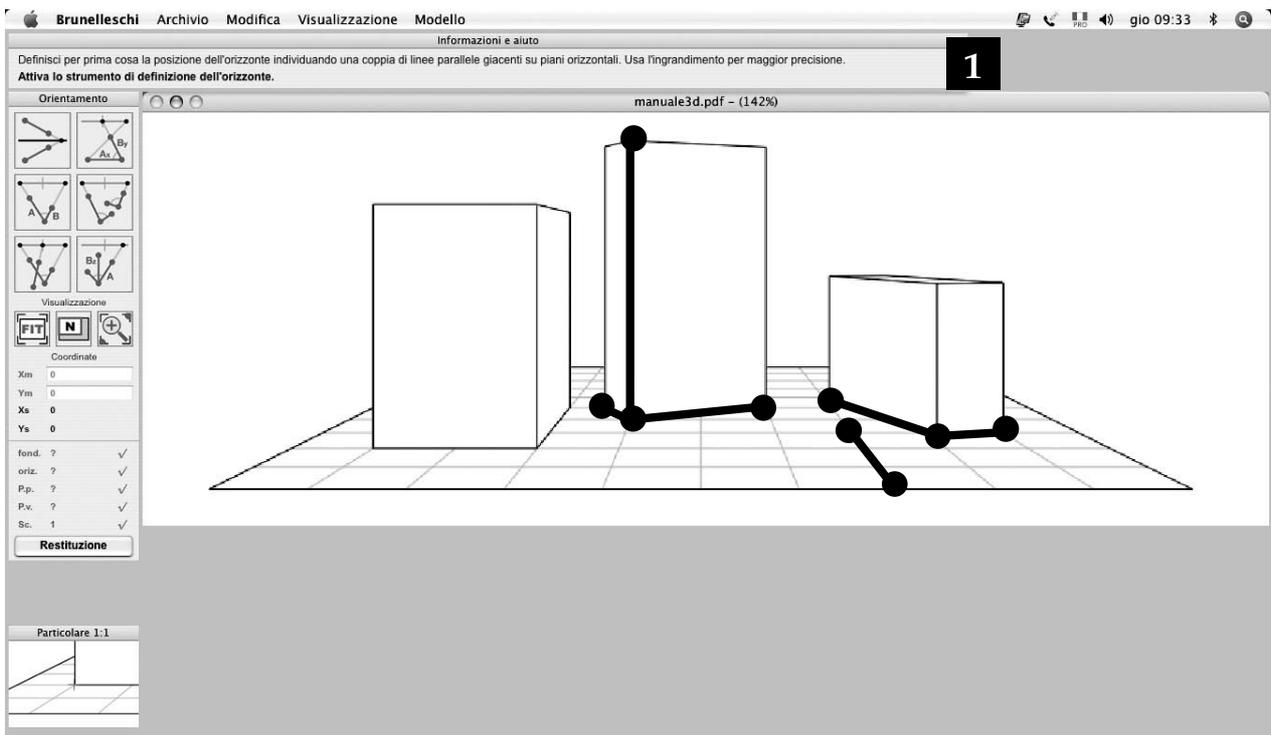
È possibile se sull'immagine prospettica è presente la deformata di un rettangolo, non ruotato, giacente su un piano verticale, di proporzioni (o dimensioni) note, e un segmento appartenente allo stesso piano orizzontale del lato inferiore del rettangolo, che con esso formi un angolo retto (fig. nella pagina successiva, al centro).

Attivato il comando il cui simbolo è mostrato qui a fianco, l'utente dovrà indicare l'angolo retto sull'immagine, attraverso tre *click* del *mouse* (il primo lato dell'angolo orizzontale retto da indicare è quello non appartenente al rettangolo succitato). Successivamente, con un ulteriore *click* del *mouse*, dovrà individuare il segmento in deformata prospettica che rappresenta l'altezza del rettangolo. Eseguita questa operazione, nella finestra di informazioni (1), verranno mostrati i campi di testo, indicati come A e B, dove sarà necessario inserire le proporzioni relative (o dimensioni reali) dei due lati; specificare, se desiderato, l'unità di misura; fare click sul bottone "assegna".



<sup>50</sup> Trattazione dell'argomento in termini geometrici in questo stesso testo, pag.41

<sup>51</sup> Come sopra, pagg.42-43



*Passaggio alla fase di definizione degli elementi geometrici della prospettiva.*

**Determinati punto di vista e punto principale, è possibile passare alla fase successiva di definizione degli elementi geometrici della prospettiva**

Eseguito uno dei comandi mostrati in precedenza, il *software* determinerà automaticamente punto di vista e punto principale. È possibile ripetere uno qualunque dei comandi descritti nelle pagine precedenti su altri elementi, se presenti e individuabili nella rappresentazione prospettica; il *software* farà la media fra le diverse coordinate calcolate, aumentando in questo modo l'accuratezza della posizione del punto di vista rispetto al quadro.

La retta fondamentale viene arbitrariamente assunta<sup>52</sup> dal software nel margine inferiore dell'immagine prospettica, e il fattore di scala corretto sarà applicato automaticamente se l'utente avrà inserito dimensioni reali coerenti quando richiesto (vedi pagg. prec.).

La determinazione di punto di vista e punto principale consente di passare alla fase di definizione degli elementi geometrici da restituire: nella *palette* degli strumenti principali di orientamento, l'utente potrà premere il bottone di comando denominato "**restituzione**".

<sup>52</sup> Trattazione dell'argomento in termini geometrici in questo stesso testo, pag.44

## DEFINIZIONE DEGLI ELEMENTI GEOMETRICI DELLA PROSPETTIVA

Quando l'utente conclude la fase di determinazione dell'orientamento della prospettiva (vedi pag. prec.), la *palette* degli strumenti di orientamento (fig.1, pag.63) viene sostituita da quella degli strumenti di restituzione (fig.2) nella medesima collocazione sullo schermo alla sinistra del foglio di lavoro.

### *Finestra degli strumenti di restituzione.*

La *palette* degli strumenti di restituzione è suddivisa in quattro sezioni; nella sezione in alto sono presenti i quindici comandi di definizione degli elementi geometrici da restituire, illustrati nelle pagine successive.

Nella sezione "Agganci e guide" è possibile, qualora occorresse, utilizzare delle "forzature" (comunemente denominate *snap*) ai comandi di restituzione, cioè fare in modo che le coordinate del punto inserito sul foglio di lavoro –con un *click* del *mouse*- in seguito all'attivazione di una forzatura, ne segua le caratteristiche relative. Le dodici forzature predisposte dal software Brunelleschi si riferiscono alle geometrie reali, e non alle deformate prospettiche; le funzioni attuabili sono descritte di seguito: **L**) coordinate libere; **Pa**) Parallelismo, al segmento di seguito selezionato con un *click* del *mouse*; **Pe**) Perpendicolarità al segmento di seguito selezionato; **X**) Forzatura della direzione x; **Y**) Forzatura della direzione y; **Z**) Forzatura della direzione z, verticale; **Es**) Aggancio a un estremo, di seguito selezionato facendo *click* col *mouse* nelle vicinanze di esso; **Al**) Appartenenza alla retta che contiene il segmento di seguito selezionato; **In**) Aggancio al punto di intersezione fra due rette che contengono i segmenti di seguito selezionati.

Le sezioni "Visualizzazione" e "Coordinate" sono analoghe a quelle della *palette* degli strumenti di orientamento (vedi pag.63).

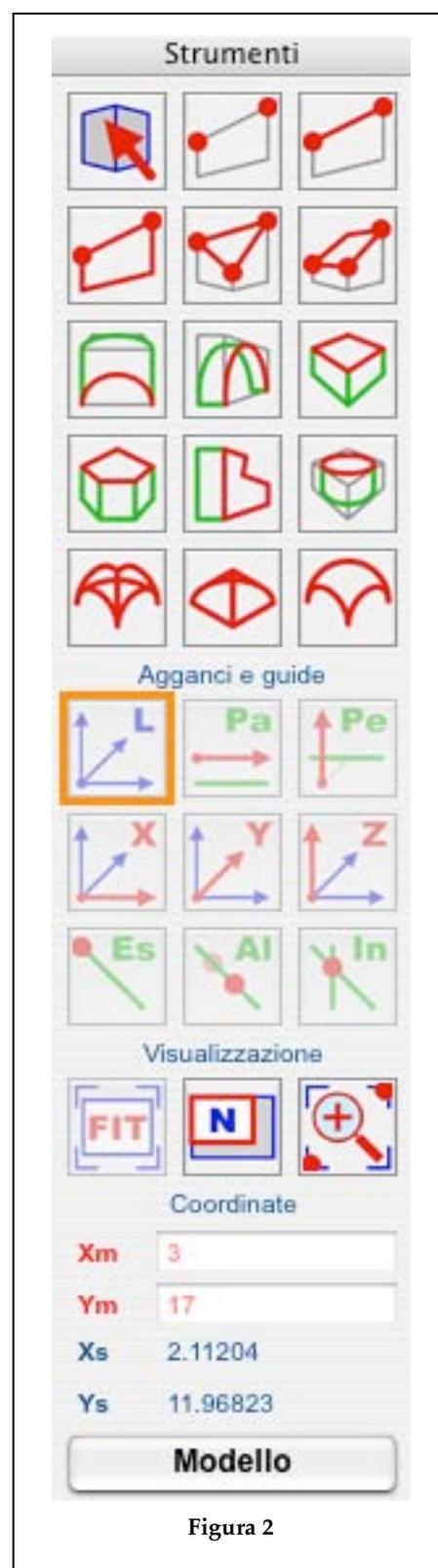
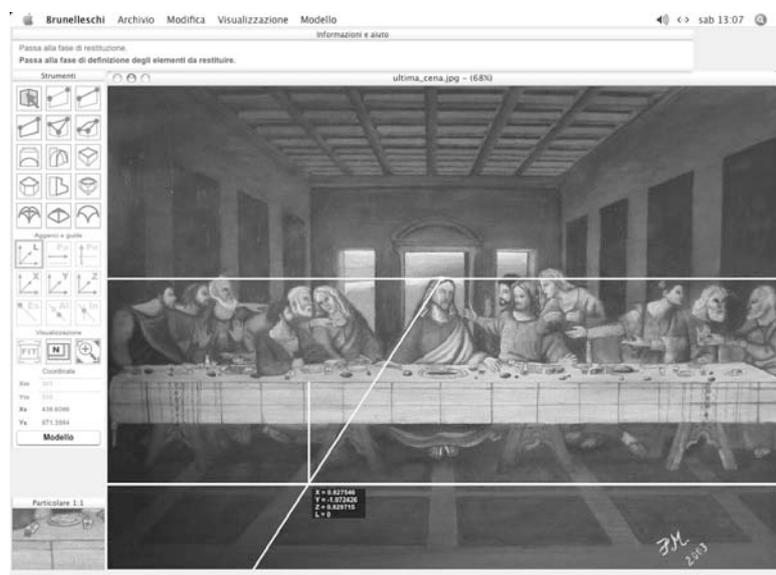


Figura 2

## METODO OPERATIVO NELLA FASE DI DEFINIZIONE DEGLI ELEMENTI GEOMETRICI DELLA PROSPETTIVA

Nel foglio di lavoro, in questa fase, ci si “muove” nello spazio deformato della prospettiva, ed è possibile definire gli elementi geometrici da restituire in vera forma

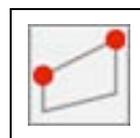
Spostando il puntatore del *mouse* sull’immagine prospettica nel foglio di lavoro, l’utente sta muovendo il “mirino” nello spazio “virtuale” della prospettiva, sul piano geometrico della stessa, come evidenziato dalla terna prospettica di assi cartesiani che ne segue il movimento; posizionato il mirino nella posizione voluta, l’utente può spostarsi in verticale tenendo premuto il tasto **option**<sup>53</sup>. Un riquadro di testo che segue il puntatore mostrerà in *real time* le coordinate reali  $x,y,z$  del punto individuato; questa semplice operazione consente un’immediato controllo preliminare della realtà metrica della prospettiva; l’informazione “L” mostra, quando applicabile, la distanza fra due punti successivi nello spazio reale. Il colore di fondo del riquadro può essere modificato premendo i tasti: V) invisibile, B) bianco, N) nero. Presa familiarità col foglio di lavoro, l’utente potrà attivare uno dei strumenti di restituzione degli elementi geometrici di seguito descritti, e in qualsiasi momento verificare le proiezioni e il modello restituito premendo il bottone “Modello” (*palette* in fig.2, pagina prec.).



<sup>53</sup> Tasto ALT su piattaforma Microsoft Windows

**Comando: segmento di riferimento.**

Consente di disegnare, sull'immagine prospettica, un segmento con qualsiasi coordinate  $x,y,z$ . Tale segmento, seppur visibile, non è restituito, pertanto tale comando permette di disegnare segmenti ausiliari di riferimento.

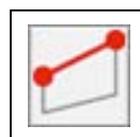


I due estremi vanno individuati cliccando due volte col *mouse* sull'immagine prospettica; si ricorda che **tenendo premuto il tasto *option***<sup>54</sup> è possibile "spostarsi di quota", in direzione verticale  $z$ ; per il punto successivo è possibile forzare la medesima quota del precedente (appartenenza del segmento ad un piano orizzontale) mantenendo premuto il tasto *shift* (maiuscole).

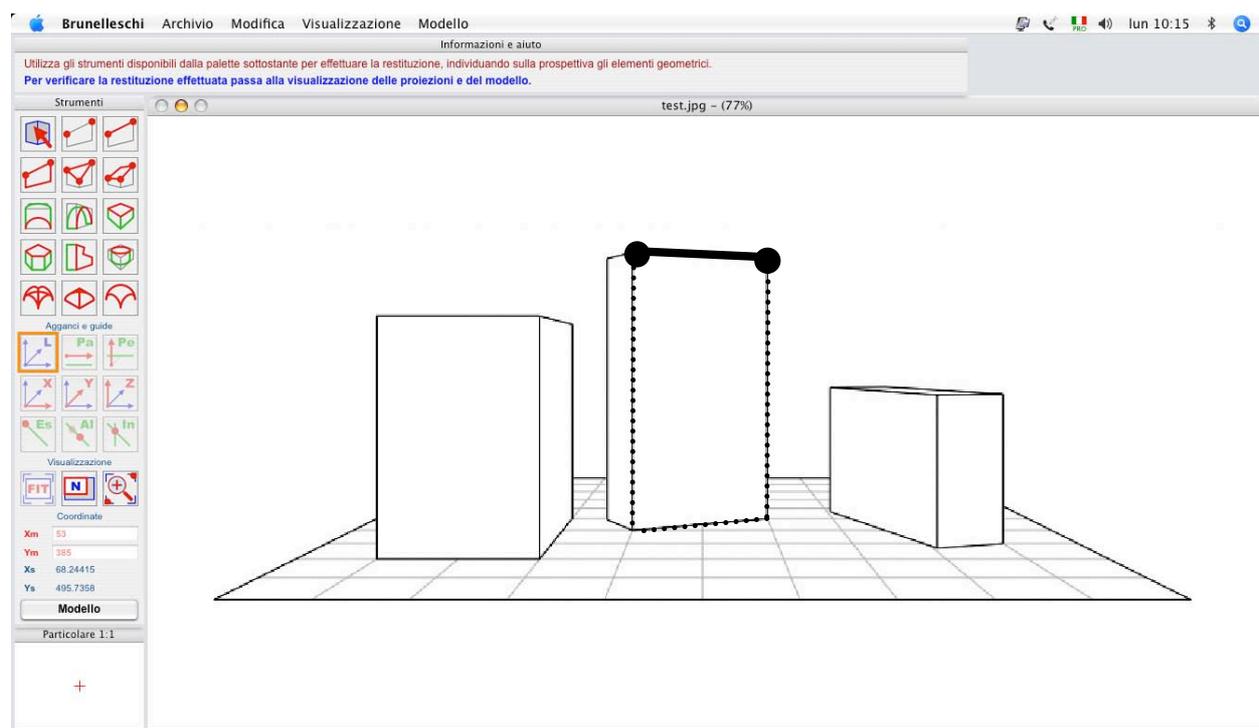
<sup>54</sup> Tasto ALT su piattaforma Microsoft Windows

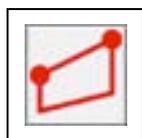
**Comando: segmento.**

Consente di disegnare, sull'immagine prospettica, un segmento con qualsiasi coordinate  $x,y,z$ . Tale segmento, contrariamente al comando descritto in precedenza, verrà restituito ed aggiunto al modello tridimensionale reale.



I due estremi vanno individuati cliccando due volte col *mouse* sull'immagine prospettica.





**Comando: piano delimitato da un segmento.**

Consente di disegnare, sull'immagine prospettica, un piano verticale delimitato: in alto, da un segmento con qualsiasi coordinate  $x,y,z$ ; in basso, dalla proiezione sul geometricale del medesimo segmento.

I due estremi vanno individuati cliccando due volte col *mouse* sull'immagine prospettica<sup>55</sup>.



**Comando: triangolo tridimensionale.**

Consente di disegnare, sull'immagine prospettica, un triangolo tridimensionale avente tre vertici nello spazio con qualsiasi coordinate  $x,y,z$ .

I tre estremi vanno individuati cliccando tre volte col *mouse* sull'immagine prospettica<sup>55</sup>.



**Comando: rettangolo e parallelogramma.**

Consente di disegnare, sull'immagine prospettica, un rettangolo tridimensionale definito da tre vertici nello spazio; i primi due potranno avere coordinate qualsiasi  $x,y,z$ , il terzo tale che sia il secondo lato del rettangolo ortogonale al primo. Consente inoltre di disegnare un parallelogramma giacente su un piano orizzontale.

I tre vertici vanno individuati cliccando tre volte col *mouse* sull'immagine prospettica<sup>55</sup>.

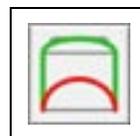
Volendo disegnare un **rettangolo**, la posizione dei primi due vertici determinerà il vincolo di posizione del terzo: se essi definiranno un segmento orizzontale, il terzo vertice potrà avere quota indipendente (vincolo di direzione ortogonale al primo segmento disegnato); se essi definiranno un segmento non orizzontale, il terzo vertice avrà la stessa quota del secondo (il rettangolo sarà giacente su un piano verticale, se lo è il primo segmento).

Volendo disegnare un **parallelogramma** giacente su un piano orizzontale, dopo aver definito i primi due vertici con la medesima quota, l'utente dovrà tenere premuto il tasto *shift* per assegnare la stessa quota al terzo punto, che potrà avere coordinate  $x,y$  indipendenti.

<sup>55</sup> Si ricorda che tenendo premuto il tasto *option* (tasto *ALT* su piattaforma Microsoft Windows) è possibile "spostarsi di quota", in direzione verticale  $z$ ; per il punto successivo è possibile forzare la medesima quota del precedente (appartenenza del segmento ad un piano orizzontale) mantenendo premuto il tasto *shift* (maiuscole)

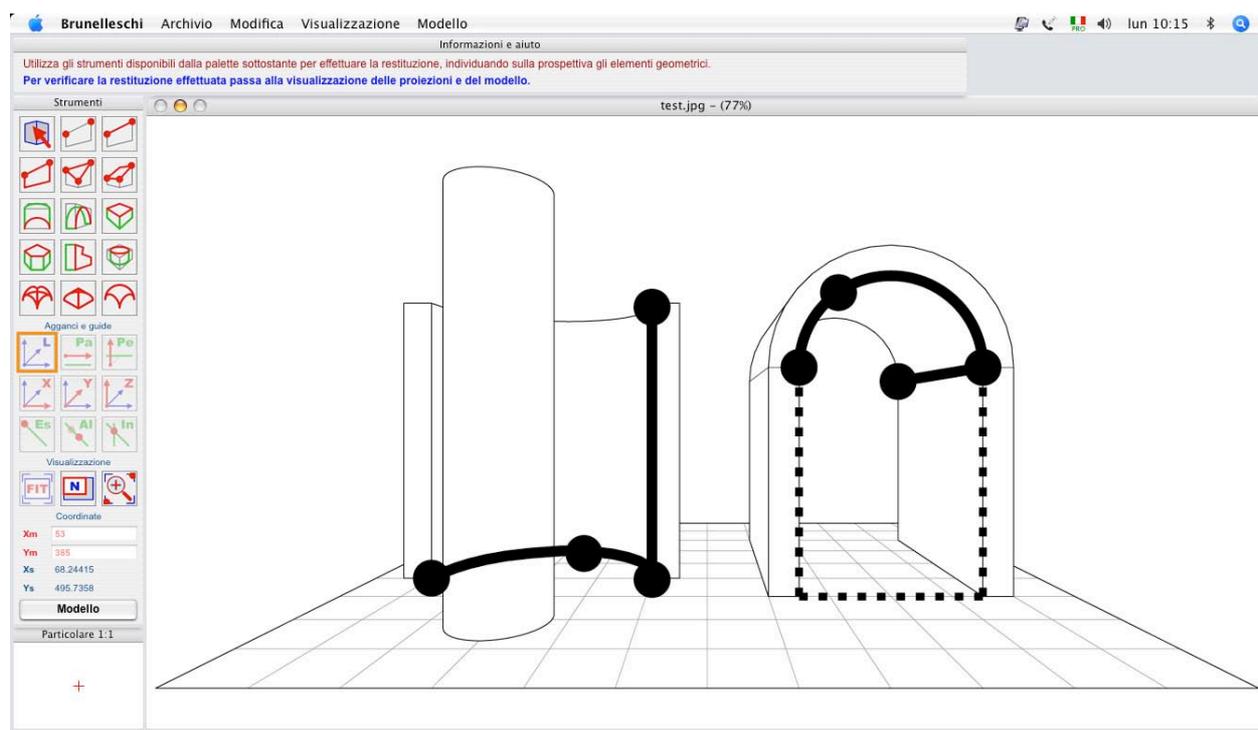
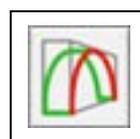
**Comando: arco appartenente ad un piano orizzontale o superficie cilindrica o conica ad asse verticale.**

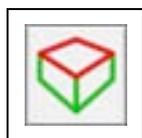
Consente di disegnare, sull'immagine prospettica, un arco giacente su un piano orizzontale, definito da tre punti di cui il primo e l'ultimo siano gli estremi, oltre alla superficie cilindrica che abbia tale arco come direttrice, e generatrice ortogonale al geometricale (es. absidi). Dopo aver digitato la risoluzione dell'arco, i tre punti vanno individuati cliccando col *mouse* sull'immagine; per generare la superficie, spostare il cursore e cliccare per definirne l'altezza (tenere premuto "option" per rastremare), o premere il tasto C per mantenere l'arco.



**Comando: arco appartenente ad un piano verticale o superficie cilindrica ad asse orizzontale o inclinato.**

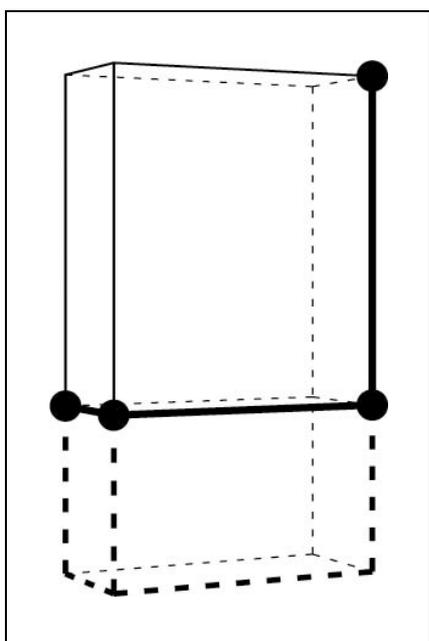
Comando analogo nell'utilizzo al precedente, consente di disegnare, sull'immagine prospettica, un arco giacente su un piano verticale, definito da tre punti di cui il primo e l'ultimo siano gli estremi; è possibile ottenere la superficie cilindrica che abbia tale arco come direttrice, e generatrice orizzontale o inclinata (es. volte a botte).





**Comando: rettangolo o parallelepipedo.**

Consente di disegnare, sull'immagine prospettica, un rettangolo giacente su un piano orizzontale, individuando tre suoi vertici, che a tale piano appartengano, e che congiunti definiscano due lati formanti fra loro un angolo retto. Consente inoltre di disegnare un parallelepipedo che abbia tale rettangolo come direttrice, e generatrice ortogonale al geometrico (vedi figura a fianco).



I tre vertici del rettangolo vanno individuati cliccando tre volte col *mouse* sull'immagine prospettica; dopo aver definito i primi due vertici -e pertanto il primo lato- l'utente potrà "forzare la quadratura" del rettangolo premendo il tasto **Q**, ossia imporre che il successivo lato sia di uguale lunghezza per ottenere un quadrato, in vera forma; il *software* imporrà comunque la direzione perpendicolare del secondo lato rispetto al primo.

Per generare il **parallelepipedo**, cliccare una quarta volta per definirne l'altezza (tenere premuto "**option**" per rastremare), altrimenti premere il tasto **C** per mantenere il solo rettangolo di base; premendo il tasto **H**, l'utente imporrà che la dimensione **z** sia di lunghezza uguale lato precedentemente inserito (utile per generare cubi).



**Comando: poligono orizzontale o prisma verticale.**

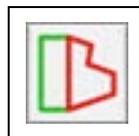
Consente di disegnare, sull'immagine prospettica, un poligono giacente su un piano orizzontale, individuando i suoi vertici, che a tale piano appartengano. Consente inoltre di disegnare un prisma che abbia tale poligono come direttrice, e generatrice ortogonale al geometrico.

I vertici del poligono vanno individuati cliccando col *mouse* sull'immagine prospettica; definito l'ultimo vertice, l'utente dovrà premere il tasto **C** per chiudere il poligono cioè verrà inserito l'ultimo lato, oppure il tasto **A** per mantenerlo aperto -ultimo lato omesso.

Per generare il **prisma**, cliccare una quarta volta per definirne l'altezza, altrimenti premere il tasto **C** per mantenere il solo poligono di base.

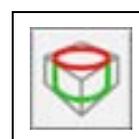
**Comando: poligono verticale o prisma orizzontale.**

Comando analogo nell'utilizzo al precedente, consente di disegnare, sull'immagine prospettica, un poligono giacente su un piano verticale, individuando i suoi vertici, che a tale piano appartengano. Consente inoltre di disegnare un prisma che abbia tale poligono come direttrice, e generatrice parallela al geometrico.



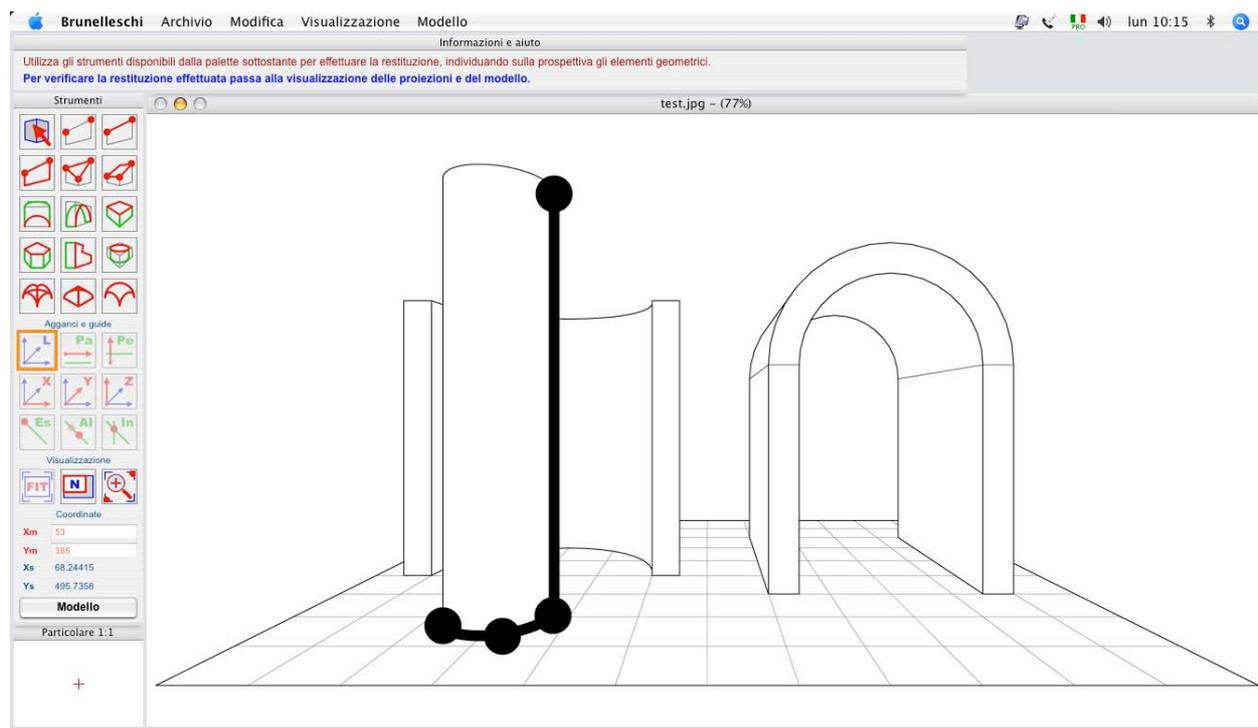
**Comando: cerchio o cilindro.**

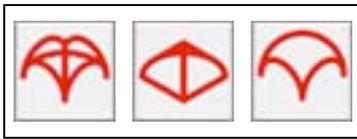
Consente di disegnare, sull'immagine prospettica, un cerchio giacente su un piano orizzontale, definito da tre punti ad esso appartenenti; è possibile ottenere il volume cilindrico che abbia tale cerchio come direttrice, e generatrice ortogonale al geometrico.



Dopo aver digitato la risoluzione (lati) del cerchio, i tre punti, che univocamente lo determinano vanno individuati cliccando col *mouse* sull'immagine prospettica.

Per generare il **cilindro**, spostare il cursore e cliccare per definirne l'altezza (tenere premuto "**option**" per rastremare), altrimenti premere il tasto **C** per mantenere il solo cerchio di base.



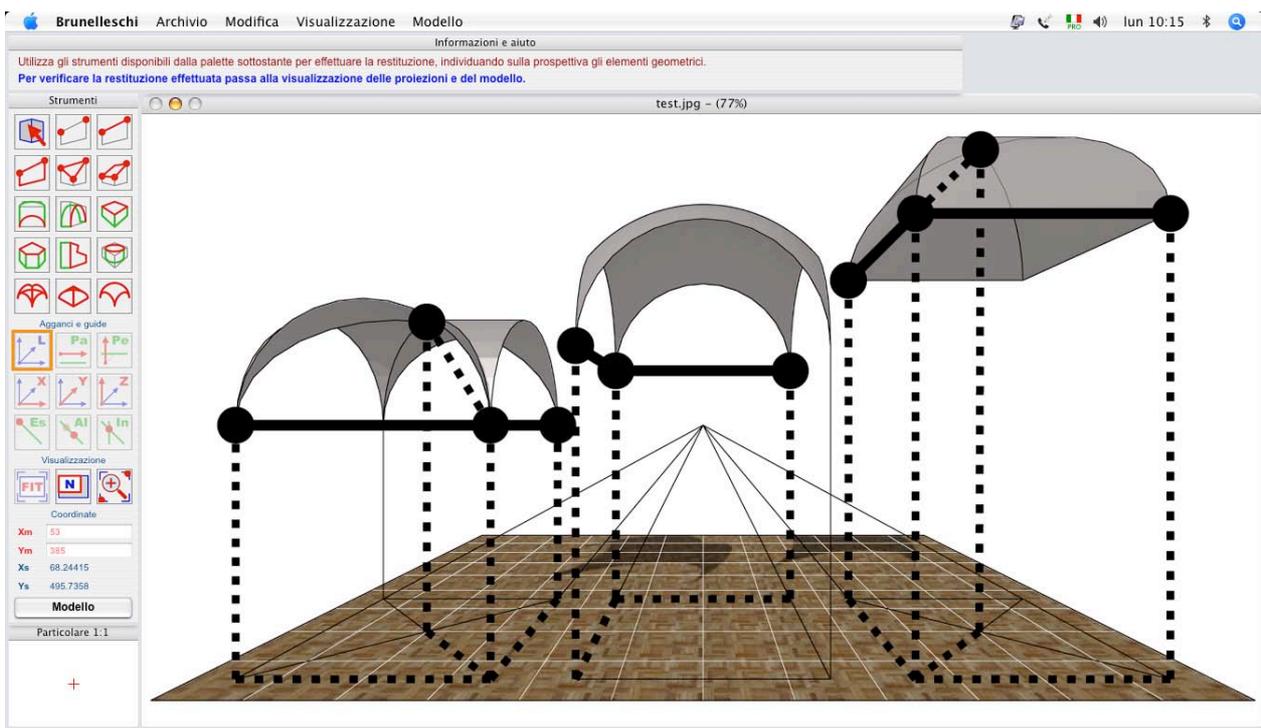


**Comandi:** *volta a crociera, padiglione, vela sferica.*

Consente di disegnare, sull'immagine prospettica, le superfici di volte a crociera, padiglione o vela sferica, individuando il rettangolo di imposta -ovvero tre dei suoi vertici appartenenti ad un piano orizzontale, tali che congiunti definiscano due lati formanti fra loro un angolo retto- e, nel caso delle volte a crociera o a padiglione, l'altezza della chiave.

I tre vertici del rettangolo d'imposta vanno individuati cliccando tre volte col *mouse* sull'immagine prospettica; dopo aver definito i primi due vertici -e pertanto il primo lato- l'utente potrà "forzare la quadratura" del rettangolo premendo il tasto **Q**, ossia imporre che il successivo lato sia di uguale lunghezza per ottenere un quadrato, in vera forma; il *software* imporrà comunque la direzione perpendicolare del secondo lato rispetto al primo.

Nel caso di volte a crociera o a padiglione, cliccare una quarta volta per definirne l'altezza, che il *software* mostrerà a partire dal punto di intersezione delle "diagonali" della proiezione in pianta della volta; nel caso di volta a vela sferica tale operazione non è necessaria.



**Comando: selezione elemento definito dall'utente.**

Tale comando consente di selezionare un elemento geometrico precedentemente individuato dall'utente, per effettuare operazioni attivabili utilizzando il menu "Modello" o le relative "scorciatoie" della tastiera: cancellazione, spostamento, duplicazione dell'elemento selezionato, come verrà illustrato successivamente.



Per selezionare un elemento, dopo aver attivato il comando il cui simbolo è raffigurato qui a fianco, è sufficiente fare *click* col *mouse* sull'elemento stesso: un cambiamento del colore (*default* dal blu al rosso) dei segmenti che lo rappresentano darà conferma dell'avvenuta selezione.

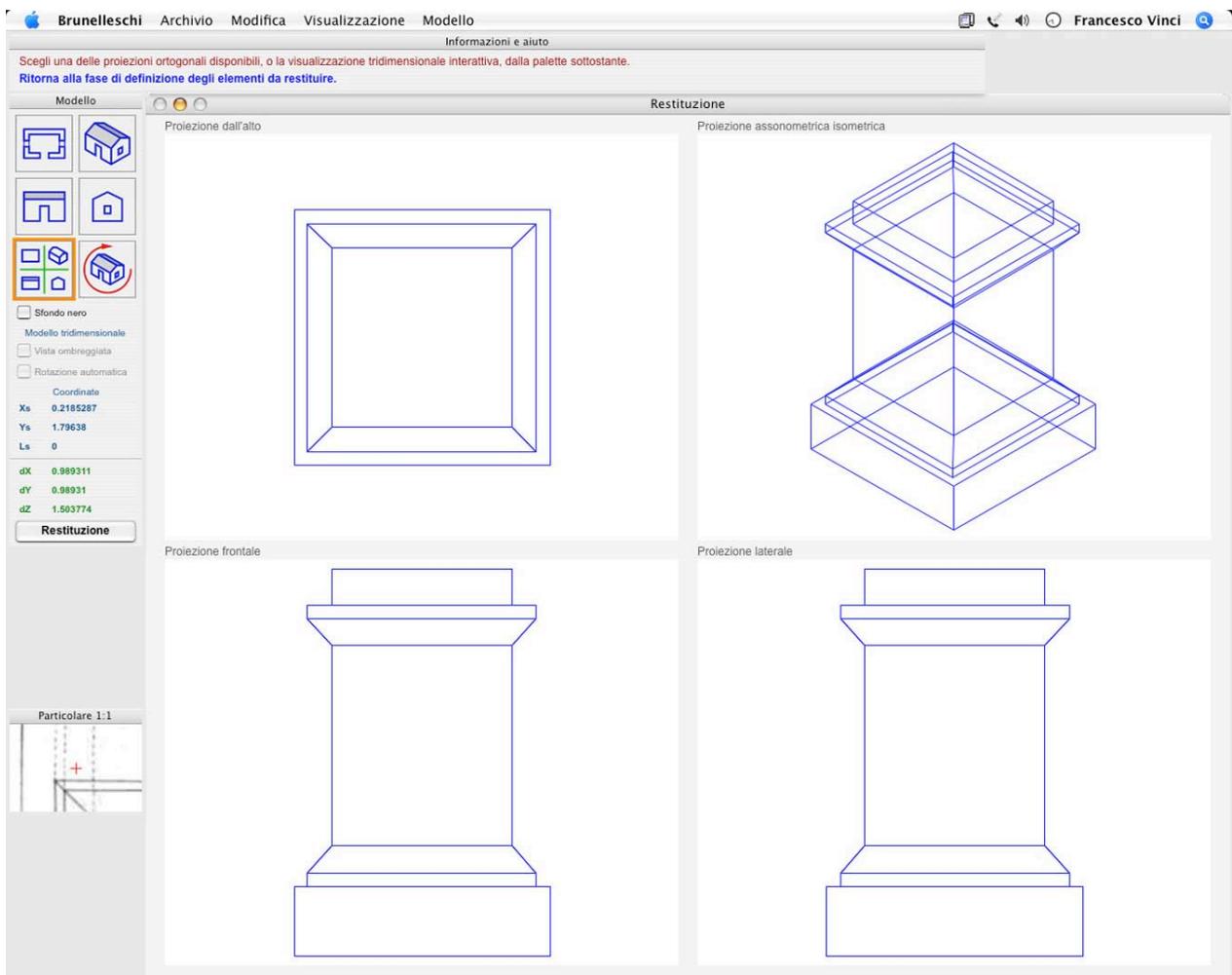
Si rimanda alla sezione di descrizione puntuale dei comandi dei menu a tendina (pag.81) per utili suggerimenti ed esemplificazioni sulla definizione di elementi ripetitivi, quali **porticati, colonnati, scale, gradonate, griglie prospettiche**. La sezione "Esempi di restituzione" (pag.xxx) allo stesso modo potrà fornire un'utile illustrazione dell'applicazione pratica degli strumenti del *software Brunelleschi*.

Ricordando quanto già accennato precedentemente, l'utente potrà in qualsiasi momento visualizzare le proiezioni e il modello restituito premendo il bottone "Modello" in basso nella *palette* degli strumenti di restituzione (fig.2, pag.69): verrà mostrata una finestra con le tre proiezioni ortogonali (proiezione dall'alto, proiezione frontale, proiezione laterale) e la proiezione assonometrica isometrica, derivate dal modello restituito. Le operazioni che è possibile eseguire sul modello restituito -misurazione di grandezze metriche, visualizzazione del modello da altri punti di vista, esportazione verso altri *software cad*, etc.- verranno illustrate nella sezione "Gestione del modello restituito", a partire dalla pagina successiva alla presente.

**L'utente potrà visualizzare le proiezioni e il modello in qualsiasi istante premendo il bottone "Modello"**

## GESTIONE DEL MODELLO RESITUITO

Se l'utente avrà premuto il bottone "Modello" (in basso nella *palette* degli strumenti di restituzione, fig.2, pag.69) il *software Brunelleschi* eseguirà il passaggio dalla fase di definizione degli elementi geometrici da restituire alla fase di gestione e controllo del modello restituito, e verrà mostrata, come detto in precedenza, una finestra con le proiezioni ortogonali (proiezione dall'alto, proiezione frontale, proiezione laterale) e quella assonometrica isometrica (figura in basso). Inoltre, la *palette* degli strumenti di restituzione sarà chiusa e sostituita da quella di visualizzazione e controllo del modello tridimensionale, le cui caratteristiche sono di seguito illustrate.



### ***Finestra di visualizzazione e controllo del modello tridimensionale restituito.***

La *palette* di visualizzazione e controllo del modello, in figura qui a fianco, è suddivisa in tre sezioni; nella sezione in alto sono presenti i sei comandi che consentono di scegliere quale rappresentazione del modello restituito dovrà essere mostrata nella finestra principale della restituzione; da sinistra a destra, e dall'alto in basso, tali comandi sono: proiezione ortogonale dall'alto, proiezione assonometrica isometrica, proiezione ortogonale frontale, proiezione ortogonale laterale, visualizzazione (di *default*) delle quattro succitate proiezioni contemporaneamente, visualizzazione del modello tridimensionale in prospettiva (vedi pag. succ.). Il comando opzionale "Sfondo nero" imposta come nero il colore del "foglio", se attivato dall'utente.

Alla sezione denominata "Modello tridimensionale" appartengono i due comandi opzionabili, relativi alla sola modalità di visualizzazione del modello tridimensionale in prospettiva (vedi quanto detto nel paragrafo precedente); "Vista ombreggiata", se attivato, visualizza il modello restituito come "solido", con colori e ombreggiature (sorgente di luce predeterminata dal *software*), altrimenti il modello viene mostrato a "fil di ferro". Il comando "Rotazione automatica", se attivato, ruota automaticamente il modello restituito senza intervento dell'utente.

La sezione delle "Coordinate", nella parte alta, consente di controllare le coordinate  $X_s, Y_s$  del puntatore (*mouse*) sulle proiezioni ortogonali del modello reale. E' possibile leggere misure lineari in  $L_s$ , definendo un segmento con due successivi *click* del *mouse*. Più in basso sono mostrate le dimensioni d'ingombro complessivo del modello restituito,  $dX, dY, dZ$ .

Premendo il bottone "**Restituzione**", in basso, il *software* riporta l'utente alla fase di definizione degli elementi geometrici da restituire.



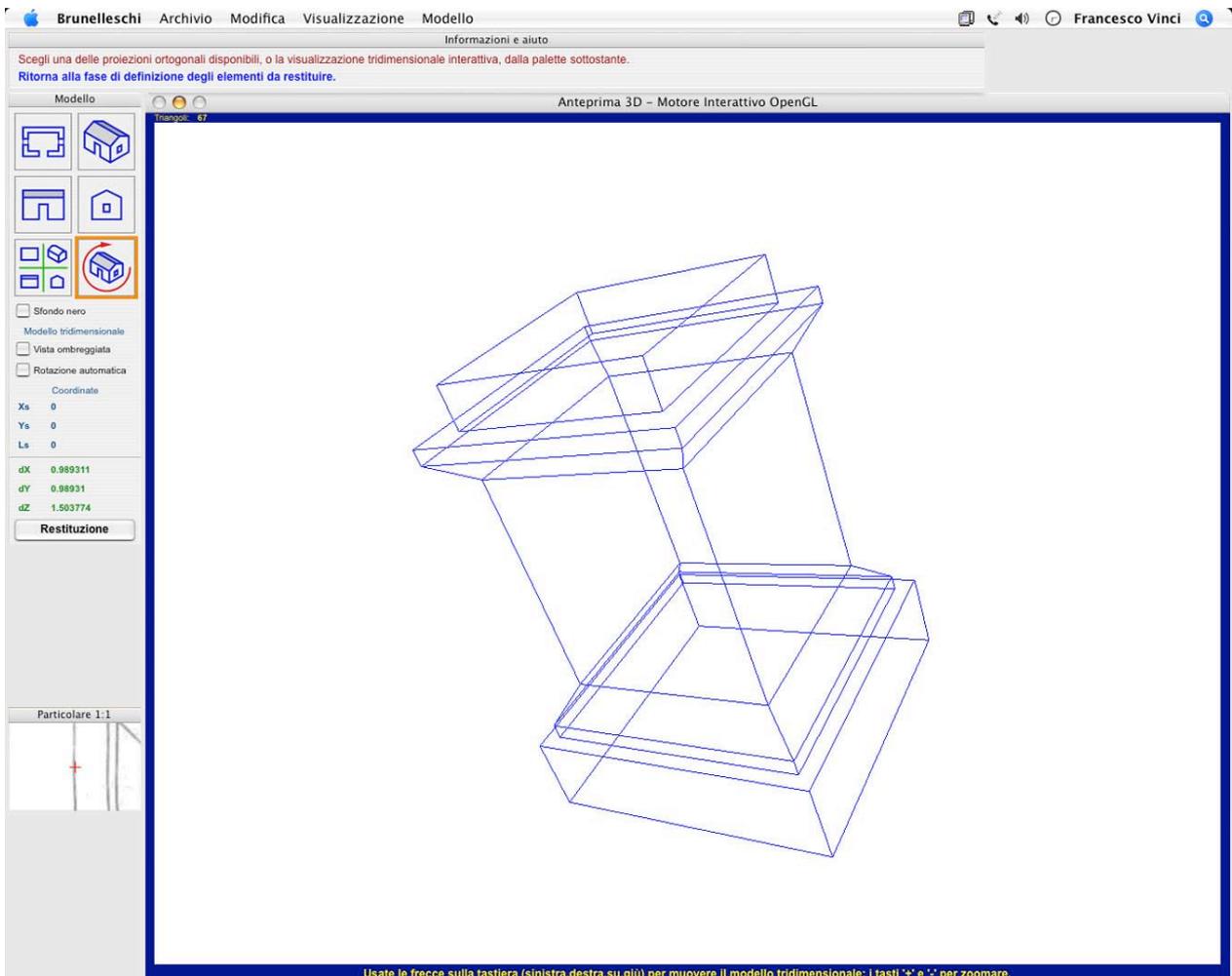
Figura 3

*Controllo della visualizzazione in prospettiva del modello tridimensionale restituito*<sup>56</sup>.

La visualizzazione in prospettiva del modello restituito, sia essa di tipo *wireframe* (fil di ferro) o *solid* (ombreggiata) può essere modificata dall'utente utilizzando la tastiera –a meno che l'utente non abbia attivato la “rotazione automatica”, come spiegato nella pagina precedente.

La “scena” –così definito l'insieme di oggetti restituiti– viene in partenza (*default*) osservata “frontalmente”, ma il punto di vista potrà “girare intorno” ad essa se l'utente premerà i tasti freccia –sinistra o destra– così come abbassarsi o alzarsi con i tasti freccia in alto o in basso. Il punto di vista potrà essere avvicinato o allontanato dalla scena premendo i tasti “+” e “-” (più e meno).

<sup>56</sup> Tale modalità funziona se sul proprio sistema operativo sono installati sia Apple QuickTime che la libreria Quesa; vedere i requisiti di sistema e la procedura di installazione a pagg.57-58



### *I menu a tendina: elenco dei comandi e descrizione delle funzioni.*

Di seguito vengono illustrati i comandi attivabili dalle voci dei menu a tendina del *software Brunelleschi*; si ricorda che su sistema Apple Mac OS X la “barra” dei menu è posizionata in alto nello schermo principale, mentre su sistema Microsoft Windows si trova nella parte superiore della finestra di informazioni di *Brunelleschi*.

#### **MENU “BRUNELLESCHI”**

##### **Informazioni su Brunelleschi..<sup>57</sup>**

Mostra una finestra di informazioni: la versione attuale del *software*, il titolo della tesi di dottorato, la data di consegna, la denominazione e il ciclo dello stesso; per chiudere la finestra, premere il bottone “OK”.

##### **Preferenze..<sup>58</sup>**

Mostra una finestra di impostazione dei parametri generali del *software Brunelleschi* personalizzabili dall’utente; è possibile impostare il colore –facendo *click* col *mouse* sul riquadro colorato- e lo spessore a video delle linee di lavoro (e selezione), orientamento, proiezione (e costruzione), restituzione (e modello) –trascinando col *mouse* gli *slider* accanto al numero che indica lo spessore, espresso in *pixel*.

È possibile impostare la profondità di colore dell’immagine prospettica da importare e restituire (valore ottimale 24 *bit*) per ottimizzare la memoria RAM disponibile; maggiore profondità di colore comporta una maggiore quantità di memoria richiesta; la profondità di 1 *bit* importa l’immagine prospettica come *bitmap* in bianco e nero (solo *pixel* bianchi o neri).

Per memorizzare e applicare la personalizzazione, premere il bottone “Imposta”; per riportare i parametri personalizzabili ai valori predefiniti da Brunelleschi (default) premere il bottone “Ripristina”; premere il bottone “Annulla” per chiudere la finestra di personalizzazione senza compiere alcuna modifica.



<sup>57</sup> Su piattaforma Microsoft Windows tale voce si trova in fondo al menu “Modello”

<sup>58</sup> Su piattaforma Microsoft Windows tale voce si trova nel menu “Archivio” sopra la voce “Esci”

## **MENU "ARCHIVIO"**

### ***Nuova restituzione..***

Con questo comando l'utente può iniziare (eliminando ogni precedente lavoro, qualora presente nel foglio di lavoro) l'esecuzione della restituzione assistita con *Brunelleschi* importando un'immagine prospettica nel foglio di lavoro.

La finestra di selezione file che apparirà a seguito della scelta di tale voce dal menu, consentirà di sfogliare il contenuto dei supporti magnetici o ottici (*hard disk, cd-rom*, o altro) su cui è memorizzata l'immagine prospettica di tipo raster -secondo uno dei formati riconoscibili dal *software Brunelleschi* attraverso Apple QuickTime- e di importarla selezionandola e premendo il bottone di comando *Apri*.

Per poter procedere con la restituzione, è necessario ottenere il codice di autorizzazione, collegato al codice di richiesta fornito da *Brunelleschi* all'atto di importazione dell'immagine, univoco per ciascun *file* di immagine (vedi pagg.61-62 per una spiegazione più estesa sulle procedure di importazione e autorizzazione).

### ***Apri..***

Consente di aprire un documento (*file*) in precedenza registrato dall'utente, nel formato proprio del *software Brunelleschi* (b3f), contenente le informazioni sulla posizione nel supporto magnetico (*hard disk* o altro) del *file* dell'immagine prospettica, gli elementi di orientamento e tutti gli elementi geometrici individuati dall'utente sulla prospettiva.

La finestra di selezione file che apparirà a seguito della scelta di tale voce dal menu, consentirà di sfogliare il contenuto dei supporti magnetici o ottici (*hard disk, cd-rom*, o altro) su cui è memorizzato il documento da aprire e di aprirlo selezionandolo e premendo il bottone di comando *Apri*, in maniera del tutto analoga a quanto possibile con la gran maggioranza degli applicativi disponibili per *personal computer*.

### **Registra**

Consente di registrare un documento (*file*) nel formato proprio del *software Brunelleschi* (b3f), contenente la posizione nel supporto magnetico (*hard disk* o altro) del *file* dell'immagine *raster*, gli elementi di orientamento e tutti gli elementi geometrici individuati sulla prospettiva.

Tale documento sovrascriverà, con lo stesso nome e nella medesima posizione, il documento attualmente aperto, se in precedenza registrato.

### **Registra come..**

Consente di registrare un documento (*file*) nel formato proprio del *software Brunelleschi* (b3f, vedi comando precedente "Registra").

La finestra di indicazione file che apparirà a seguito della scelta di tale voce dal menu, consentirà di sfogliare il contenuto dei supporti magnetici o ottici (*hard disk*, *cd-rom*, o altro) per scegliere la posizione in cui memorizzare il documento, premendo il bottone di comando **Registra**, in maniera del tutto analoga a quanto possibile con la gran maggioranza degli applicativi per *personal computer*.

### **Esporta..**

Consente di registrare un *file* nei formati ASCII DXF (Autodesk) o GDL (Graphisoft ArchiCAD) per consentire post-elaborazioni con altri *software*; dal sotto-menu l'utente potrà selezionare una delle seguenti voci:

*DXF 2D Proiezioni ortogonali*, proiezioni di Monge e assonometria isometrica dal modello restituito.

*DXF 3D Filo di ferro*, modello tridimensionale a "filo di ferro", solo spigoli e contorni, senza solidi e superfici.

*DXF 3D Solidi e superfici*, modello tridimensionale completo con solidi e superfici.

*GDL 3D Filo di ferro*, modello tridimensionale a "filo di ferro", solo spigoli e contorni, senza solidi e superfici.

*GDL 3D Solidi e superfici*, modello tridimensionale completo con solidi e superfici.

### **Esci**

Chiude l'applicativo *Brunelleschi*.

**È possibile registrare il modello restituito o le sue proiezioni attraverso la voce "Esporta.." dal menu "Archivio"**

### ***MENU "MODIFICA"***

Il menu "Modifica" ha un funzionamento del tutto analogo a quello presente nella gran maggioranza degli applicativi disponibili per *personal computer*.

#### ***Annulla***

Annulla l'ultima operazione effettuata, ripristinando lo stato precedente alla sua esecuzione (quando possibile).

#### ***Taglia***

In un campo di testo, cancella e memorizza la parte selezionata dall'utente.

#### ***Copia***

In un campo di testo, memorizza la parte selezionata dall'utente.

#### ***Incolla***

In un campo di testo, scrive il testo in precedenza memorizzato dall'utente.

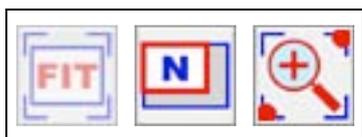
#### ***Cancella***

In un campo di testo, cancella la parte selezionata.

### ***MENU "VISUALIZZAZIONE"***

#### ***Vista ottimizzata***

Voce che attiva il comando "FIT", mostra l'immagine in modo tale che appaia integralmente sullo schermo (vedi particolare della *palette* degli strumenti di orientamento o di restituzione qui a fianco, descrizione a pag.63 e pag.69).



#### ***Ingrandimento***

Voce che attiva il comando "+", consente di ingrandire una porzione dell'immagine indicando su di essa un rettangolo contenente la parte da visualizzare, cliccando due volte per definirne gli estremi della diagonale.

#### ***Dimensione reale***

Voce che attiva il comando "N", mostra l'immagine (o parte di essa, per immagini di grande risoluzione) in *pixel* reali, cioè in modo tale che il rapporto *image pixels/monitor pixels* sia di 1:1 (questo è il fattore di ingrandimento ideale di lavorare sull'immagine in *Brunelleschi*).

### ***Mostra elementi con costruzione***

Nella fase di definizione degli elementi geometrici da restituire, mostra gli elementi geometrici con proiezioni ed eventuali linee di costruzione, che possono essere utili come riferimenti per successivi elementi da individuare.

### ***Mostra elementi senza costruzione***

Nella fase di definizione degli elementi geometrici da restituire, mostra gli elementi senza proiezioni ed eventuali linee di costruzione.

### ***Nascondi elementi***

Nella fase di definizione degli elementi geometrici da restituire, nasconde tutti gli elementi definiti; può essere utile in fase avanzata di lavoro, in presenza di molti elementi in precedenza individuati dall'utente.

### ***Assi cartesiani di riferimento***

Se attivato, nella fase di definizione degli elementi geometrici da restituire, mostra una terna prospettica di assi cartesiani di riferimento, che segue coordinate del puntatore (cursore del *mouse*) sul foglio di lavoro, oltre al *box* di testo che mostra in *real time* le coordinate reali  $x,y,z$  del punto individuato; si ricorda, come detto in precedenza, che il colore di fondo del *box* di testo può essere modificato premendo i tasti: V) invisibile, B) bianco, N) nero.

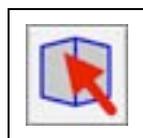
### ***Particolare a grandezza reale***

Se attivato, la finestra del particolare a grandezza reale (in rapporto 1:1 fra *image pixels/monitor pixels*; vedi pag.60) viene aggiornata in *real time* (con un ritardo minimo, di circa un secondo) seguendo le coordinate del puntatore (cursore del *mouse*) sul foglio di lavoro. Se disattivo, la finestra rimane visibile ma non aggiornata.

### ***Statistiche***

Mostra una finestra con alcune informazioni, fra cui la coordinata  $y$  della retta di orizzonte, le coordinate  $x$  e  $y$  del centro di vista, il fattore di scala, oltre al numero dei segmenti prospettici definiti sull'immagine e restituiti nel modello tridimensionale reale.

### MENU "MODELLO"

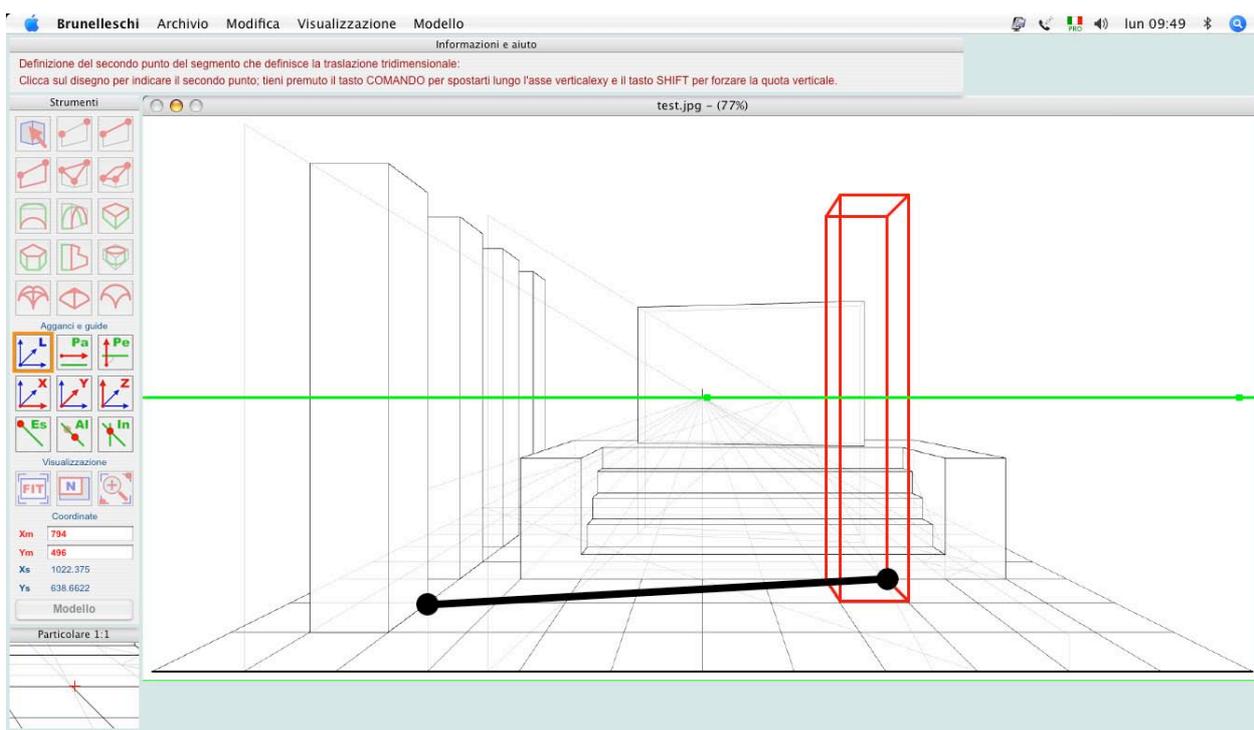


Il menu "Modello" consente di compiere le operazioni di seguito illustrate, alcune delle quali disponibili a seguito della selezione di un elemento geometrico in precedenza definito, utilizzando il comando dalla *palette* degli strumenti di restituzione il cui simbolo è mostrato qui a fianco (vedi pag.77).

#### Muovi elemento

Consente di spostare l'elemento selezionato (vedi quanto scritto sopra, o leggi pag.77) definendo un vettore di traslazione con coordinate tridimensionali libere o vincolate (vedi figura in basso).

I due estremi del vettore di traslazione vanno individuati cliccando due volte col *mouse* sull'immagine prospettica; tali estremi possono essere liberi e svincolati dall'oggetto da traslare, così come "ancorati" ad uno dei suoi vertici (o a vertici di un altro elemento geometrico definito in precedenza dall'utente sulla prospettiva), utilizzando, dalla *palette* degli strumenti di restituzione, uno dei comandi di "forzatura" precedentemente illustrati (vedi pag.69).

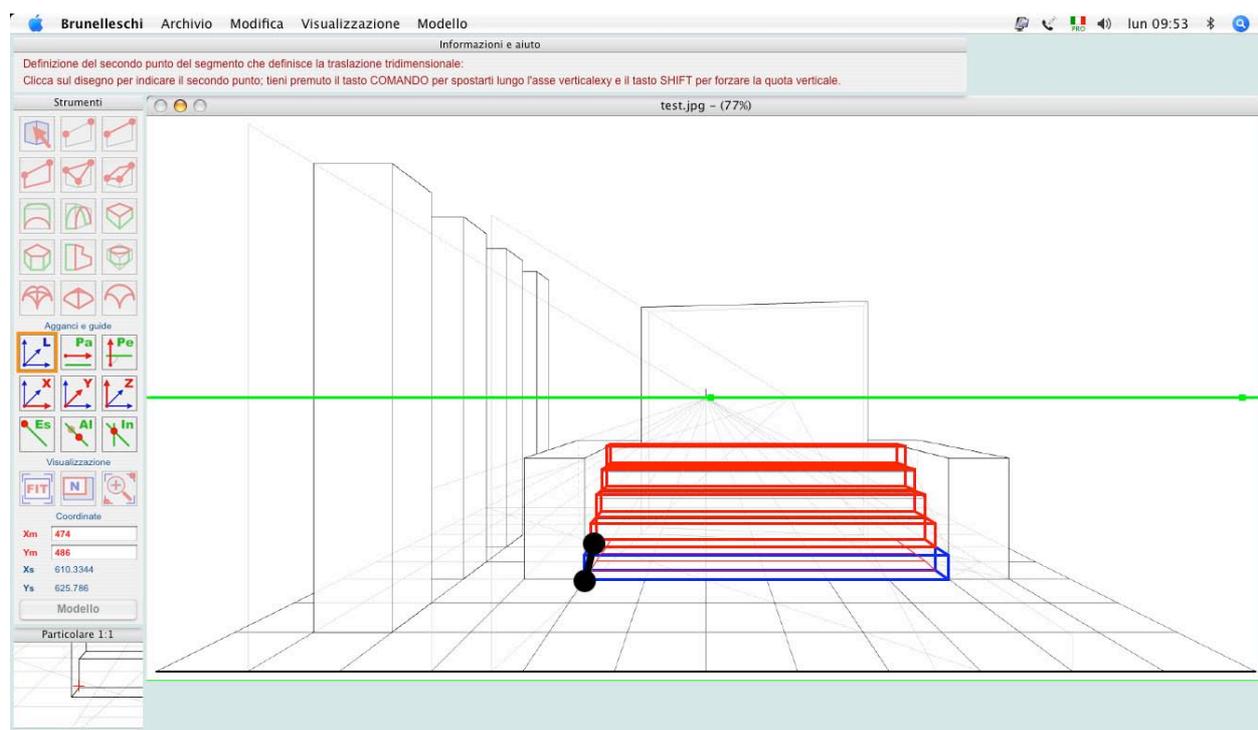


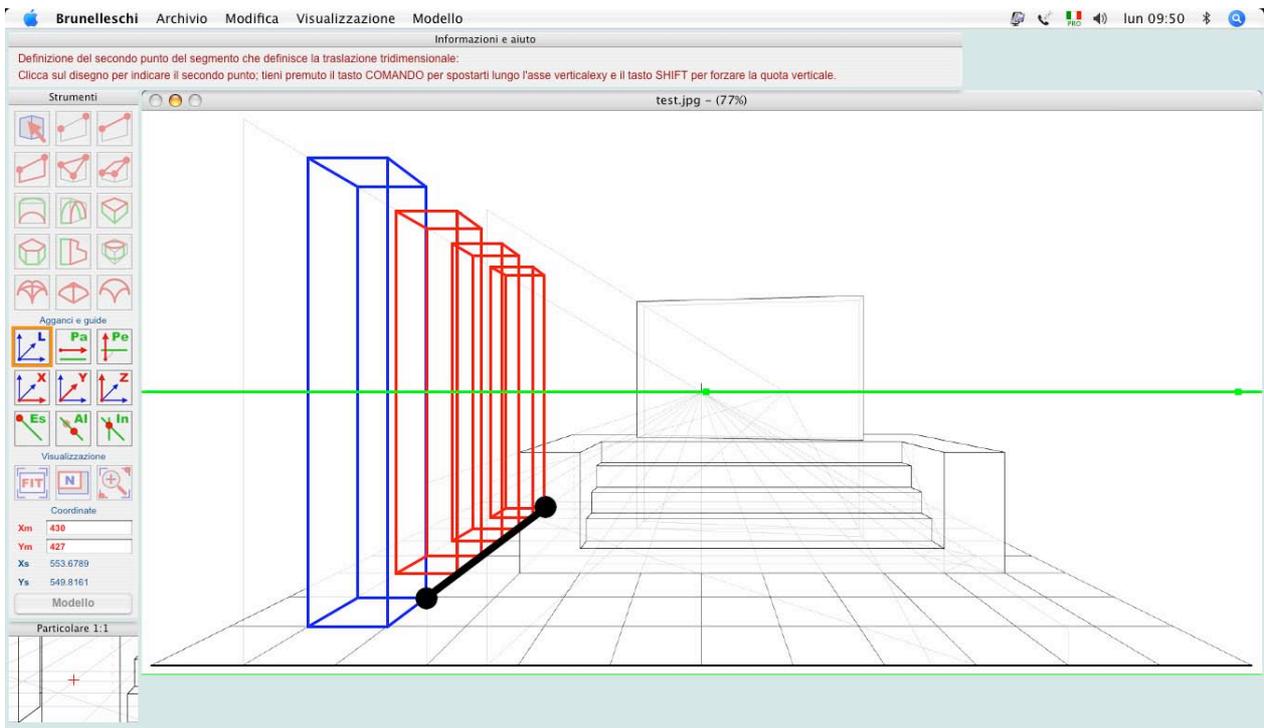
### ***Duplica elemento***

Consente di duplicare l'elemento selezionato definendo un vettore di traslazione con coordinate tridimensionali libere o vincolate.

Nella finestra di informazioni l'utente dovrà specificare il numero di copie e il metodo di duplicazione (incremento o distribuzione), confermando la scelta con l'attivazione del bottone "duplica" in succitata finestra. Successivamente, i due estremi del vettore di traslazione vanno individuati cliccando due volte col *mouse* sull'immagine prospettica –sono utilizzabili le opzioni di forzatura di cui alla voce "Muovi elemento" in precedenza illustrata.

Le modalità di duplicazione disponibili sono state definite "incremento" o "distribuzione"; la modalità "**incremento**" richiede la definizione del vettore di traslazione fra il primo elemento della serie (quello originale selezionato dall'utente) e il successivo; le copie ulteriori verranno ciascuna traslata secondo lo stesso vettore rispetto alla precedente (vedi figura in basso, esempio di duplicazione di gradoni).





La modalità “**distribuzione**” richiede la definizione del vettore di traslazione fra il primo elemento della serie (quello originale selezionato dall’utente) e l’ultimo; le copie intermedie verranno ciascuna traslata di una frazione del vettore quello definito dall’utente secondo il numero di copie impostato (vedi figura in alto, esempio di duplicazione di pilastris).

***Elimina elemento***

Elimina l’elemento selezionato.

***Crea griglia***

Se l’elemento selezionato è un rettangolo (o quadrato), consente di disegnare una griglia secondo il numero di maglie indicate dall’utente nella finestra di informazioni.

***Elimina linee sovrapposte***

Elimina eventuali linee sovrapposte.

***Elimina orientamento***

Elimina l’orientamento attuale della prospettiva e ritorna alla fase di definizione dello stesso (vedi pag.63).

***Elimina tutti gli elementi***

Elimina tutti gli elementi geometrici da restituire definiti dall’utente presenti sul foglio di lavoro.

### ***Scorciatoie da tastiera e tips.***

Alcuni comandi da tastiera, disponibili in alcune fasi della restituzione, non sono stati citati nelle pagine precedenti, sebbene possano risultare utili in determinati frangenti.

Premendo il tasto **O** (lettera), quando coerente, l'utente imporrà che la quota del successivo punto da inserire sia zero (sul geometrale).

Durante l'utilizzo dei comandi di definizione degli elementi da restituire, i bottoni di ingrandimento o riduzione dell'immagine nella *palette* relativa non sono attivi, ma l'utente può utilizzare i tasti asterisco, "+" o "-", in congiunzione con il puntamento del *mouse*, per visualizzare l'ingrandimento di una porzione della immagine o il ripristino della visualizzazione intera.

E' possibile lavorare in modalità *fullscreen* e attivare i comandi col menu contestuale, premendo il tasto **F**.

La forzatura di aggancio all'estremo **Es** funziona in modo tale che l'utente può effettuare un *click* col *mouse* nei pressi di uno dei due estremi di un segmento per agganciarlo, o nei pressi del centro dello stesso per agganciare il punto medio "reale"; nel caso di definizione dell'altezza (generatrice) di solidi prismatici, l'aggancio all'estremo viene eseguito ignorando le coordinate x,y ed assegnando la quota di aggancio z dell'estremo (utile per creare solidi di altezza uguale).

È possibile inserire le coordinate numericamente, o più convenientemente spostarsi con esattezza di un *pixel* "reale" dell'immagine *raster*, indipendentemente dal fattore di ingrandimento visualizzato, utilizzando i tasti **freccia**, e il tasto **invio** per assegnare il punto; la precisione del punto determinato può essere controllata nella finestra di visualizzazione del particolare (in basso a sinistra).

**È possibile spostarsi con precisione utilizzando i tasti freccia e assegnare il punto con il tasto "invio"**

### *Comandi in modalità beta (sperimentali).*

Alcune funzionalità di *Brunelleschi* sono disponibili in modalità sperimentale, ossia con limitazioni di funzionamento e senza documentazione.

Sono presenti i comandi da menu: **esporta: dxf 2d prospettiva** (esporta la prospettiva di riferimento vettorializzata dall'utente); **modello: rivoluzione superficie** (crea una superficie di rivoluzione selezionando successivamente una polilinea complanare e un segmento asse di rivoluzione, parallelo a due piani di riferimento cartesiani); **modello: rivoluzione duplicati** (crea duplicati per rotazione selezionando successivamente un oggetto e un segmento asse di rivoluzione, parallelo a due piani di riferimento cartesiani).

**È possibile definire in modo manuale la scala di output della restituzione tenendo premuto il tasto `command` (o `windows`) e facendo click col mouse su un segmento di cui è nota la dimensione reale**

L'utente può visualizzare una finestra di informazioni su un elemento definito cliccando su di esso tenendo premuto il tasto "**command**" (tasto "**windows**"); in tale finestra è possibile leggere la lunghezza reale del segmento scelto, impostare il valore stesso, se noto, e pertanto la scala di *output* della restituzione, oltre alla possibilità di scelta di mostrare o nascondere ciascuno degli elementi definiti dall'utente in precedenza, selezionando i bottoni di opzione dall'elenco mostrato nella parte bassa.

### *Known bugs.*

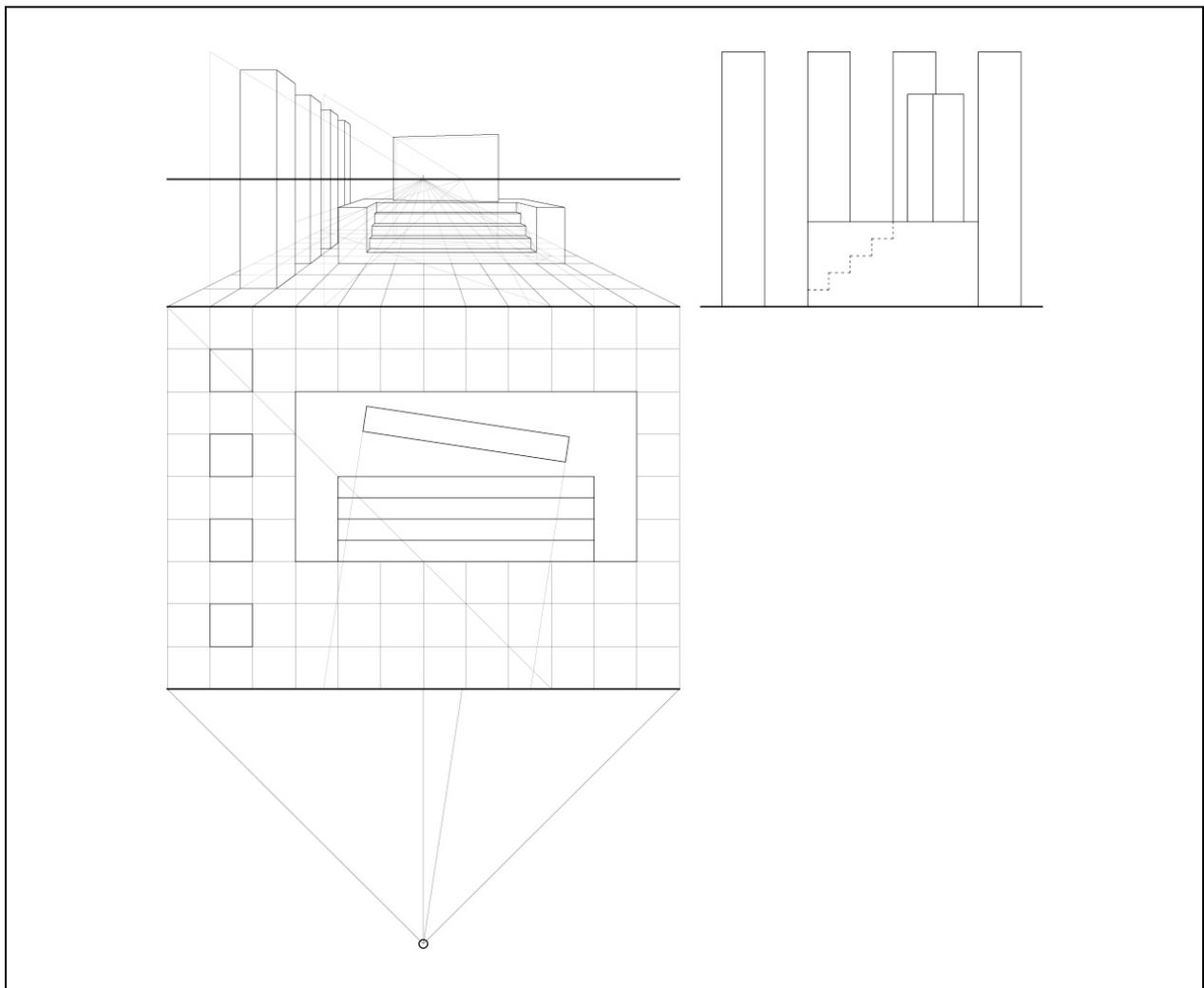
In alcune fasi di lavoro l'immagine da restituire può sparire dallo sfondo completamente o in parte; per ovviare a tale inconveniente (dovuto a problemi con alcune schede video) l'utente può uscire dal comando in corso (tasto **ESC**) e attivare una delle opzioni di ingrandimento / riduzione / ottimizzazione dell'immagine prospettica, o utilizzare i tasti "+" o "-" come spiegato in precedenza.

*ESEMPI DI RESTITUZIONE*

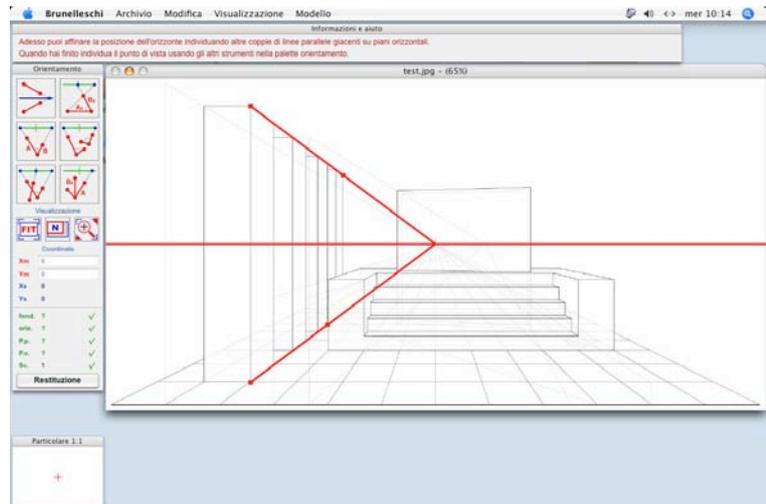


RESTITUZIONE DA UNA RAPPRESENTAZIONE  
PROSPETTICA

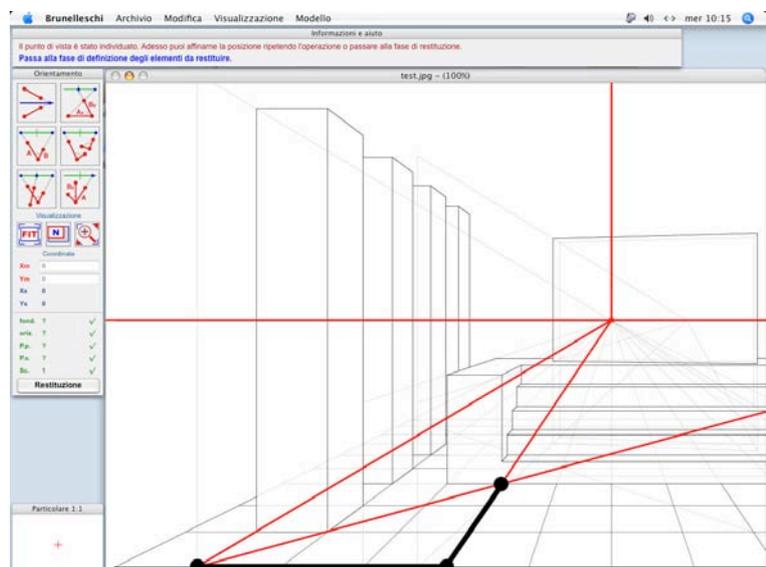
La seguente dimostrazione di restituzione con l'utilizzo del *software Brunelleschi* viene effettuata su una rappresentazione prospettica eseguita in maniera tradizionale, secondo i procedimenti noti di geometria descrittiva, con l'ausilio di un software cad per il solo tracciamento del disegno e al solo scopo di ottenere una rappresentazione esatta. Nella figura sottostante, pianta, prospetto e rappresentazione prospettica degli oggetti da restituire, con indicazione del punto di vista e della distanza principale.



Importata la rappresentazione prospettica, in formato *raster*, nell'ambiente di lavoro del software *Brunelleschi*<sup>59</sup>, si individua sulla prospettiva una coppia di rette orizzontali appartenenti allo stesso piano, dopo aver selezionato l'apposito comando<sup>60</sup>; il software ne trova il punto di fuga e determina la posizione dell'orizzonte.



L'individuazione del punto di vista è ottenuta dato un rettangolo orizzontale di dimensioni note, con due lati paralleli al quadro; scelto il comando relativo<sup>61</sup>, si individuano due lati consecutivi del rettangolo e se ne assegnano le dimensioni; il software determina gli elementi di orientamento necessari per la restituzione.



<sup>59</sup> Importazione dell'immagine secondo le modalità descritte nelle pagg.61-62

<sup>60</sup> Trattazione geometrica a pag.37, uso del software a pag.64

<sup>61</sup> Trattazione geometrica a pag.38, uso del software a pag.65

La fase di restituzione consente, spostando il puntatore del *mouse*, di muoversi nello spazio tridimensionale della rappresentazione prospettica, controllando in *real time* le coordinate "reali" degli oggetti raffigurati<sup>62</sup>.

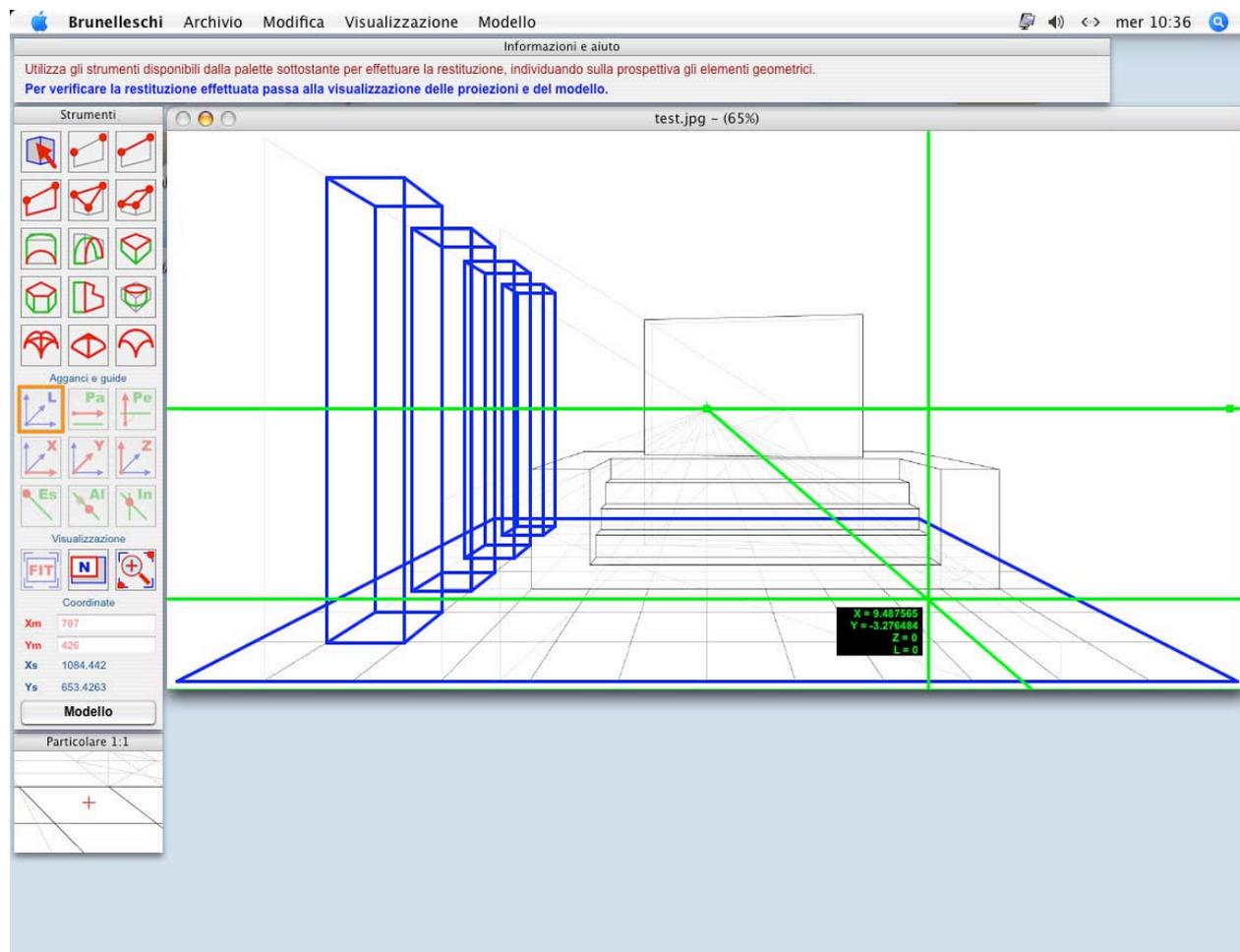
I comandi disponibili del *software Brunelleschi* consentono di restituire gli elementi rappresentati; per restituire la fila di pilastri alla sinistra dell'immagine, si seleziona lo strumento **rettangolo/parallelepipedo**<sup>63</sup> e si individuano il piano di base e il primo dei quattro plinti, attivando l'opzione di "forzatura" della direzione **X**, così come la forzatura della "quadratura" della base del plinto<sup>64</sup>; i successivi tre plinti sono determinati con l'uso dello strumento di selezione e duplicazione, in modalità distributiva<sup>65</sup>, selezionando il plinto e indicando il vettore di traslazione: il primo punto del vettore sarà agganciato a un vertice del plinto attivando la forzatura **Es**.

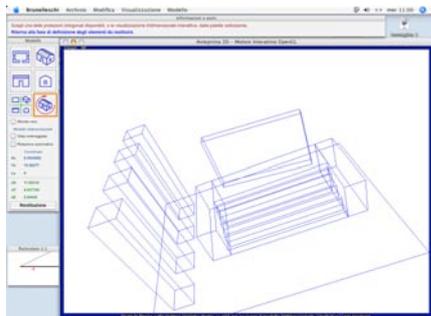
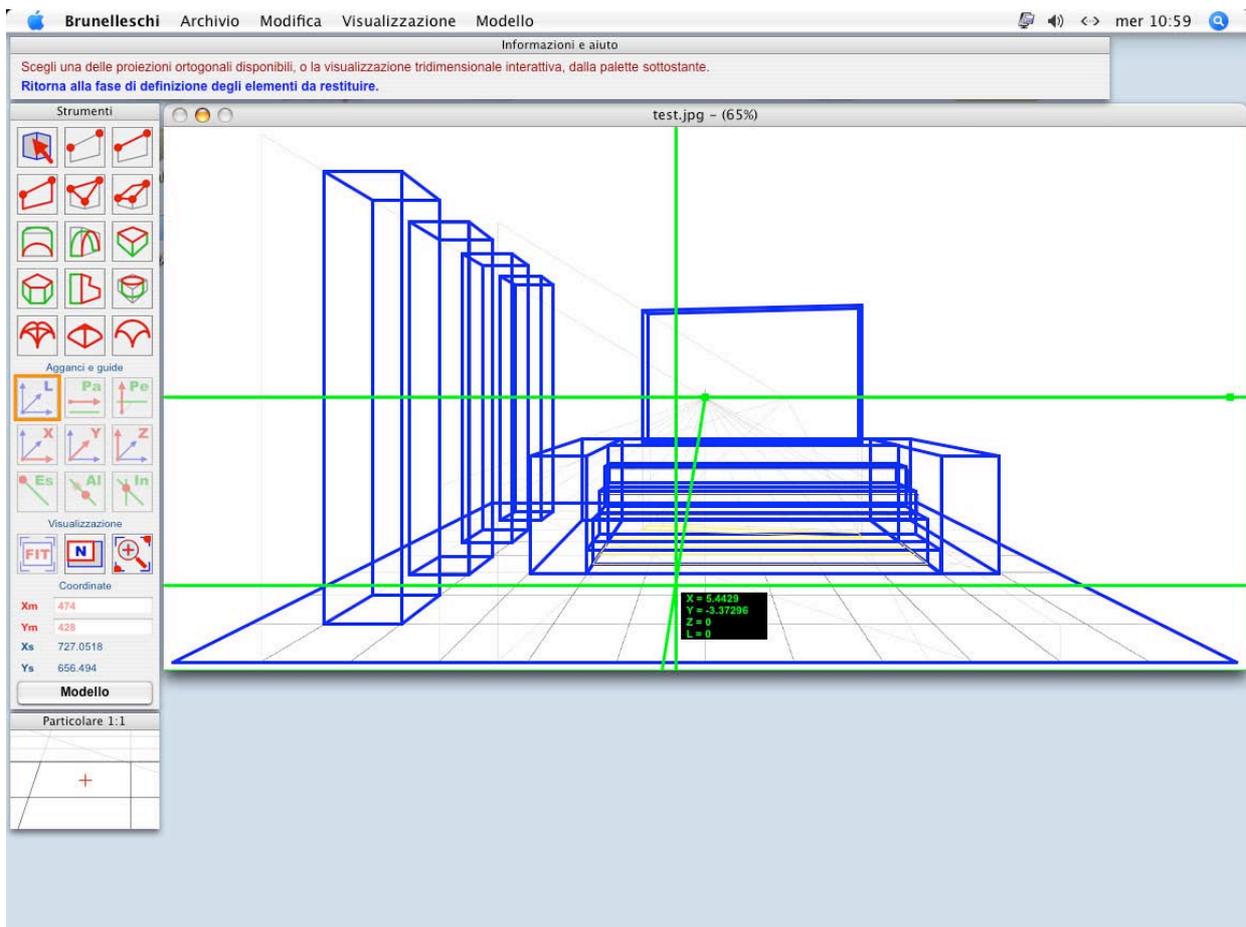
<sup>62</sup> Vedi pag.70

<sup>63</sup> Uso del comando illustrato a pag.74

<sup>64</sup> Tipi di "forzature", pag.69

<sup>65</sup> Duplicazione di elementi, pag.88





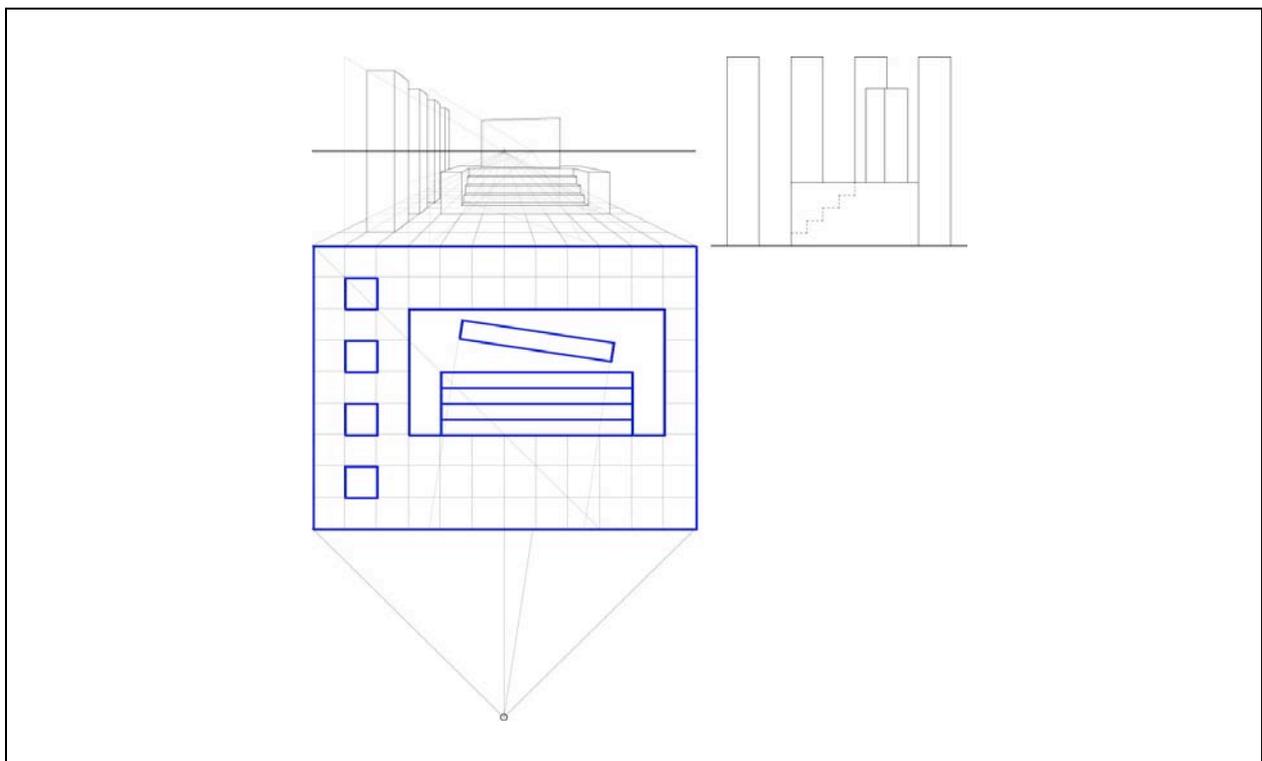
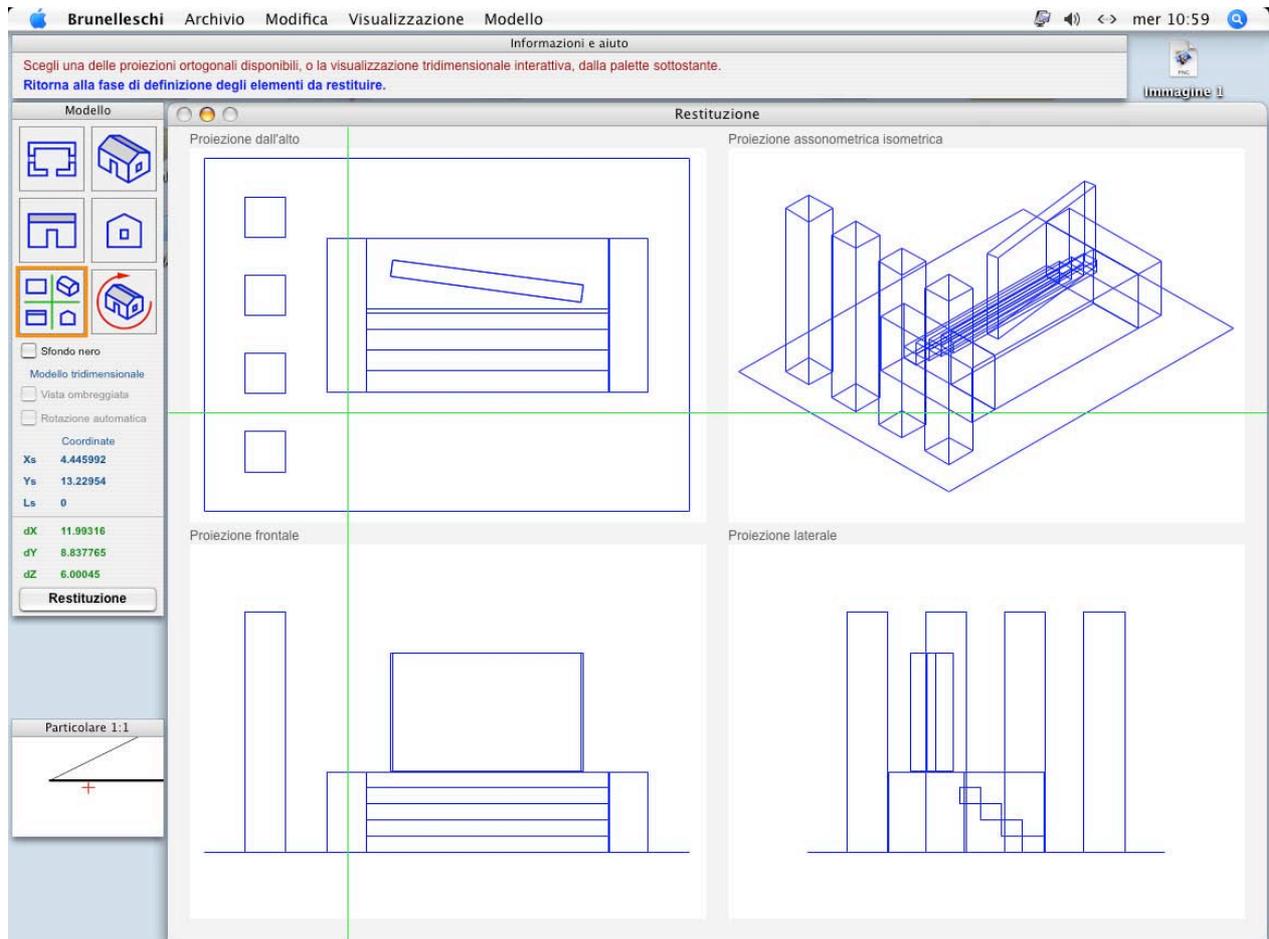
L'individuazione degli oggetti da restituire viene completata aggiungendo il piano di base, l'elemento centrale con la gradonata, e il parallelepipedo (ruotato rispetto al quadro) ad esso sovrapposto.

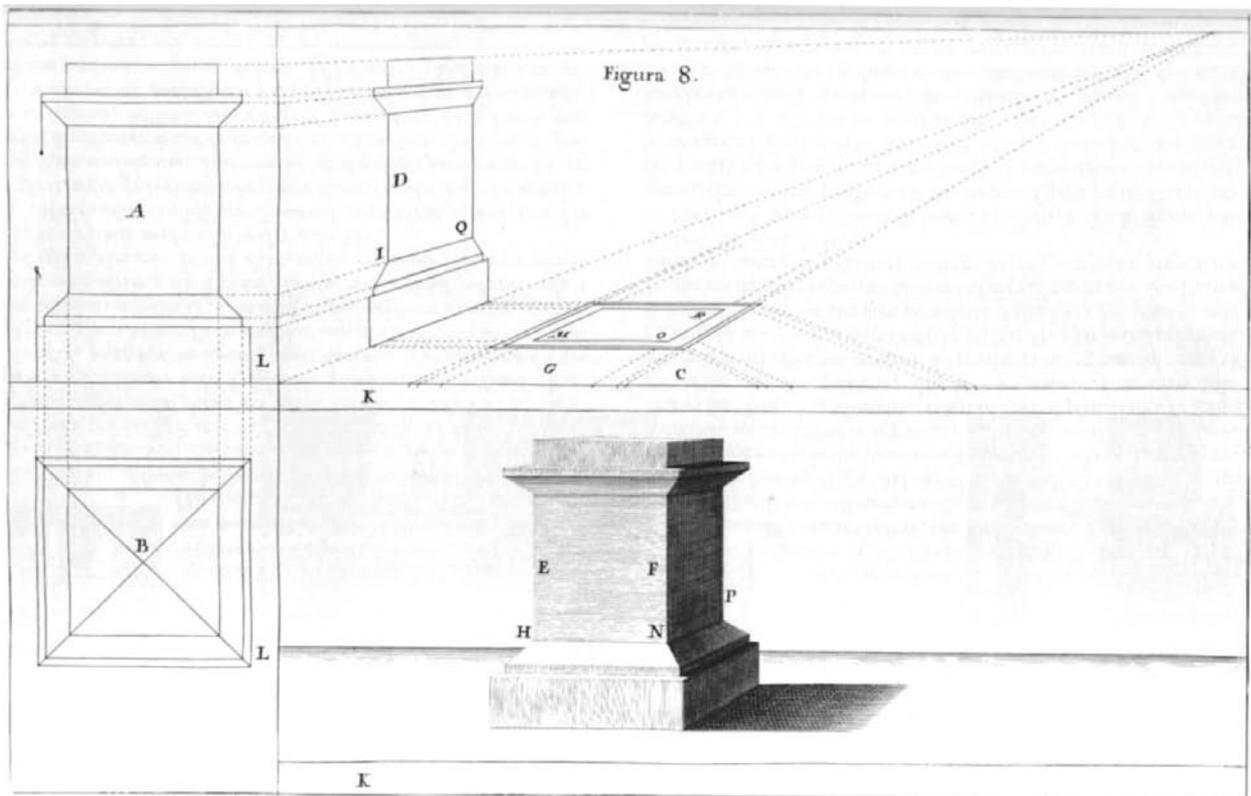
Il modello tridimensionale, da cui le proiezioni di Monge e l'assonometria in figura in alto nella pagina successiva, viene costruito e aggiornato, in tempo reale, man mano che si procede all'individuazione degli elementi da restituire; il *software Brunelleschi* consente di verificare in ogni momento il risultato del processo di restituzione<sup>66</sup>.

Il modello tridimensionale restituito può essere utilizzato per ottenere altre rappresentazioni prospettiche con semplici comandi di modifica del punto di vista<sup>37</sup>.

La sovrapposizione della pianta restituita alla pianta originaria conferma l'esattezza dell'operazione di restituzione effettuata (pag. successiva).

<sup>66</sup> Visualizzazione del modello restituito a pagg.78-80





## FIGURA OCTAVA.

*Optica projectio stylobatæ.*

**S**I libitum fuerit delineare stylobatam cum projecturis in summo & imo, incipies ab elevatione geometricâ A, ducendo occultas ad id necessarias tum versus perpendicularem L, tum deorsum pro vestigio geometrico B, cujus distantia transferentur in spatium G. Si mensuræ longitudinis distent spatium C a mensuris latitudinis, vestigium deformatum, videbitur distare a lineâ K plani, quantum est idem spatium C. In construendâ opticâ elevatione D, visuales ex punctis lineæ L dabunt lineas latitudinis; lineas verò altitudinis accipies ex lineis vestigii contracti, ut in figurâ. In formando stylobatâ nitido EF, locum anguli H dabit concursus latitudinis ex lineâ L usque ad M, & altitudinis ex lineâ K usque ad I. Concursus tum ejusdem altitudinis, tum latitudinis ex L usque ad O, dabit angulum N. Demum altitudinem anguli P accipies ex K usque ad Q; latitudinem ex L usque ad R:

## FIGURA OTTAVA.

*Piedestallo in prospettiva.*

**V**OLENDO fare un piedestallo che in cima e in fondo habbia qualche rifalto, comincerete dall' elevation geometrica A, tirando le linee occulte necessarie verso la perpendicolare L, e le altre abbasso per far la pianta geometrica B, le cui distanze doveranno portarsi nello spazio G. E se le misure della lunghezza saran lontane da quelle della larghezza lo spazio C, la pianta in prospettiva sembrerà lontana dalla linea K del piano, quanto è lo spazio C. Nel far in prospettiva l' elevatione D, le visuali tirate dalla perpendicolare L vi daranno le linee della larghezza; e quelle dell' altezza le prenderete dalla pianta in prospettiva. Nel cavare il piedestallo pulito EF, troverete il luogo preciso dell' angolo H, prendendo l' altezza dalla linea K fino a I, e la larghezza dalla linea L fino a M; poiche dove s' incontrano le due misure, quivi è l' angolo H. Per l' angolo N mantenendo la medesima altezza, la larghezza farà da L fino a O. L' altezza dell' angolo P la prenderete da K fino a Q; la larghezza dalla linea L fino a R.

## RESTITUZIONE DA UNA PROSPETTIVA

DI ANDREA POZZO

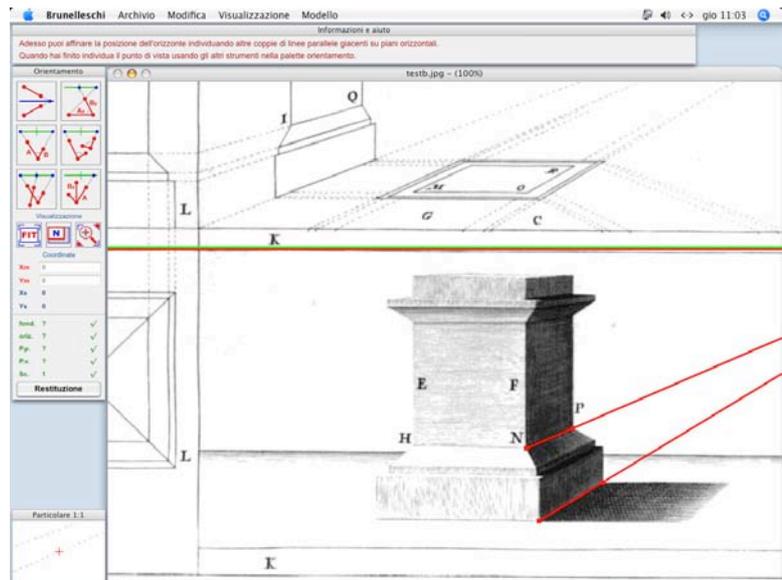
La seguente dimostrazione di restituzione con l'utilizzo del *software Brunelleschi* viene effettuata sulla rappresentazione prospettica illustrata in figura ottava dal trattato *Prospettiva de' Pittori e Architetti* di Andrea Pozzo (1642-1709), riprodotta in alto nella pagina precedente (immagine tratta dalla pubblicazione a cura di Maria Walcher Casotti, edita a Trieste, 2003).

Pubblicato per la prima volta a Roma nel 1693, il trattato è costituito da una sequenza di procedimenti pratici per "mettere in prospettiva" oggetti di difficoltà crescente, a partire da semplici volumi, agli ordini architettonici del Vignola, ad edifici di varia complessità; vi si trovano esempi di prospettive "di sotto in su" e descrizioni illustrate sul disegno e la collocazione delle scene teatrali; ciascun disegno è accompagnato da una doppia didascalia, in latino e italiano, che ne spiega il procedimento grafico; quella relativa alla figura ottava è riprodotta in basso nella pagina precedente.

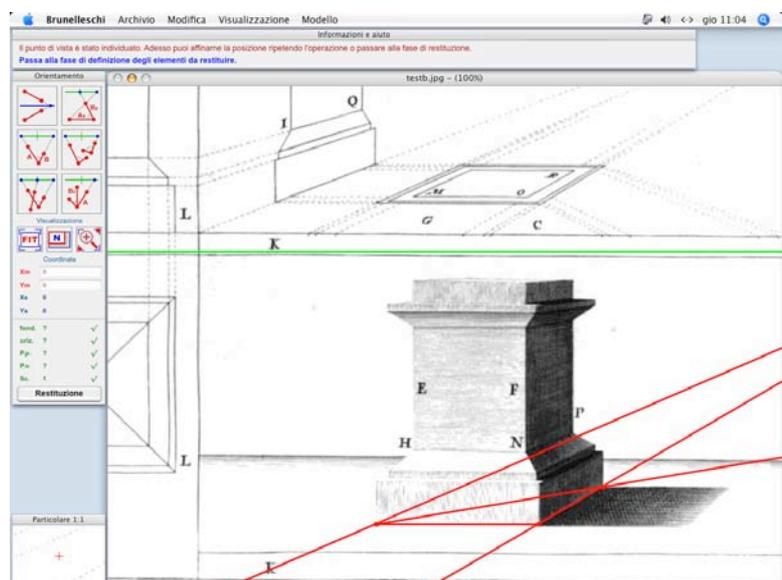
La figura ottava rappresenta un "piedistallo" che "in cima e in fondo abbia qualche risalto", in proiezione ortogonale di pianta e alzato, e in prospettiva ombreggiata; al di sopra di questa, le deformate prospettive della pianta e dell'alzato svelano il procedimento di intersezione con cui essa stessa viene costruita; ciascuna parte del disegno è contrassegnata da una lettera capitale a cui si fa riferimento nella didascalia.



Digitalizzata l'immagine utilizzando uno scanner, e importato il file nell'ambiente di lavoro di Brunelleschi<sup>67</sup>, si individuano sulla prospettiva diverse coppie di rette parallele orizzontali, attivando l'apposito comando<sup>68</sup>; il software ne trova i punti di fuga e determina l'orizzonte.



L'individuazione del punto di vista è ottenuta dato il quadrato di base di dimensioni note; scelto il comando relativo<sup>69</sup>, si individuano due lati consecutivi del quadrato e se ne assegnano le dimensioni; il software determina gli elementi di orientamento necessari la restituzione.



<sup>67</sup> Importazione dell'immagine secondo le modalità descritte nelle pagg.61-62

<sup>68</sup> Trattazione geometrica a pag.37, uso del software a pag.64

<sup>69</sup> Trattazione geometrica a pag.38, uso del software a pag.65

La fase di restituzione consente, spostando il puntatore del *mouse*, di muoversi nello spazio tridimensionale della rappresentazione prospettica, nella porzione della figura ad essa relativa, controllando in *real time* le coordinate "reali" degli oggetti raffigurati<sup>70</sup>.

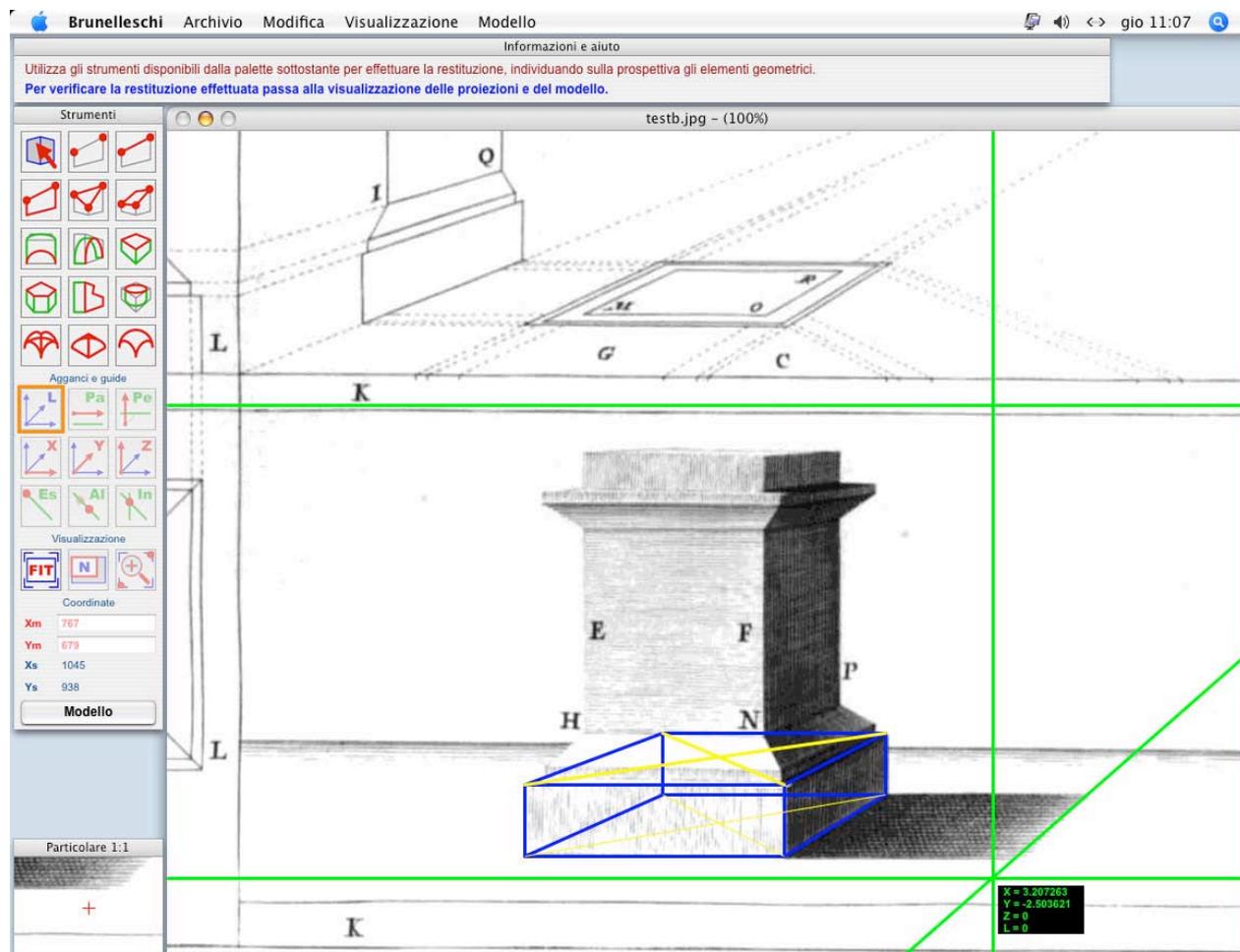
I comandi disponibili del *software Brunelleschi* consentono di restituire il piedistallo ricomponendolo dalle varie parti che lo costituiscono; utilizzando lo strumento **rettangolo/parallelepipedo**<sup>71</sup> si individua il basamento del piedistallo, attivando l'opzione di "forzatura" della direzione X, così come la forzatura della "quadratura" della base<sup>72</sup>; vengono tracciate anche le diagonali della faccia superiore del basamento, che saranno utili come "guida" per il tracciamento delle altre parti, utilizzando il comando **segmento di riferimento**<sup>73</sup> (elemento visibile ma non restituito).

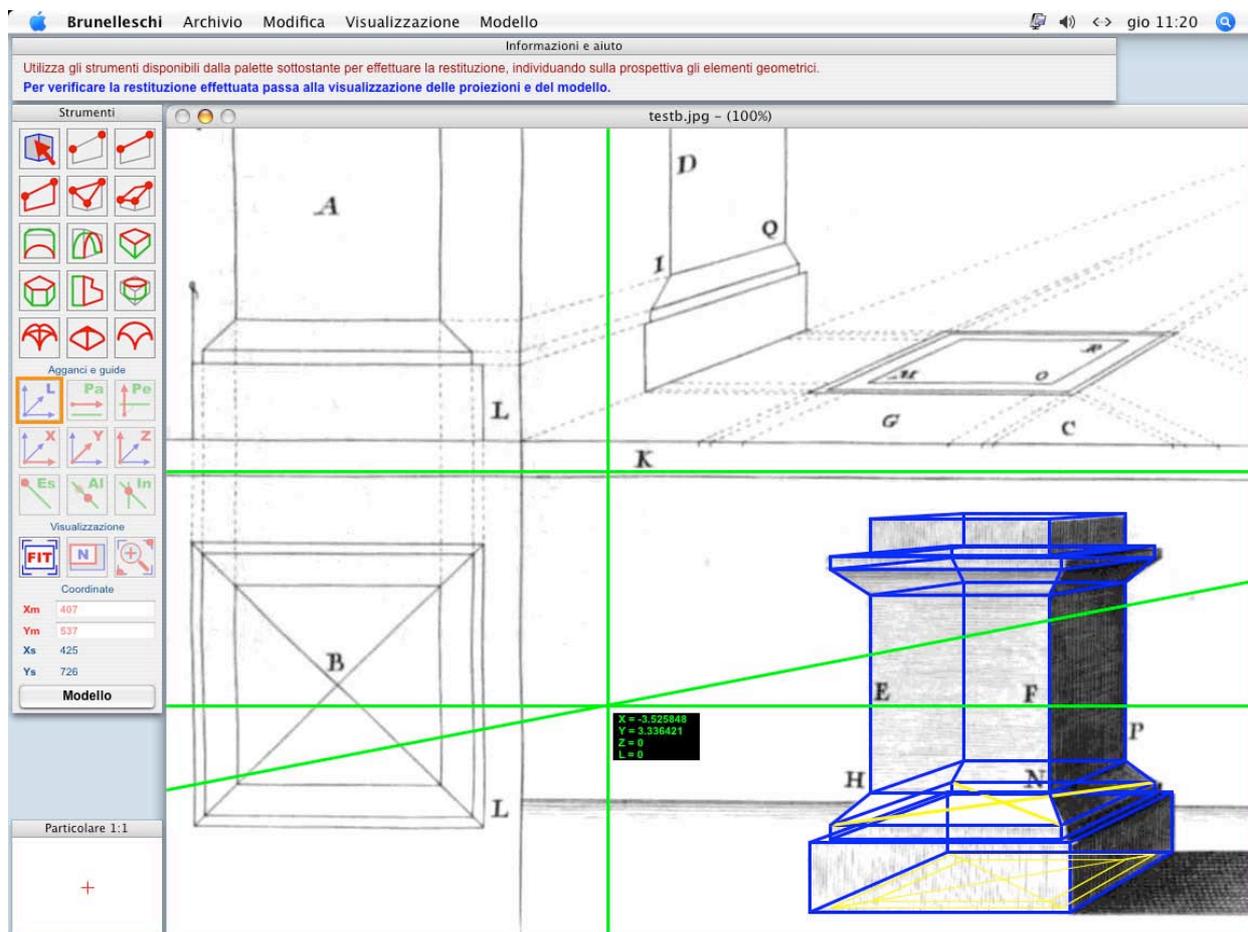
<sup>70</sup> Vedi pag.70

<sup>71</sup> Uso del comando illustrato a pag.74

<sup>72</sup> Tipi di "forzature", pag.69

<sup>73</sup> Uso del comando illustrato a pag.71





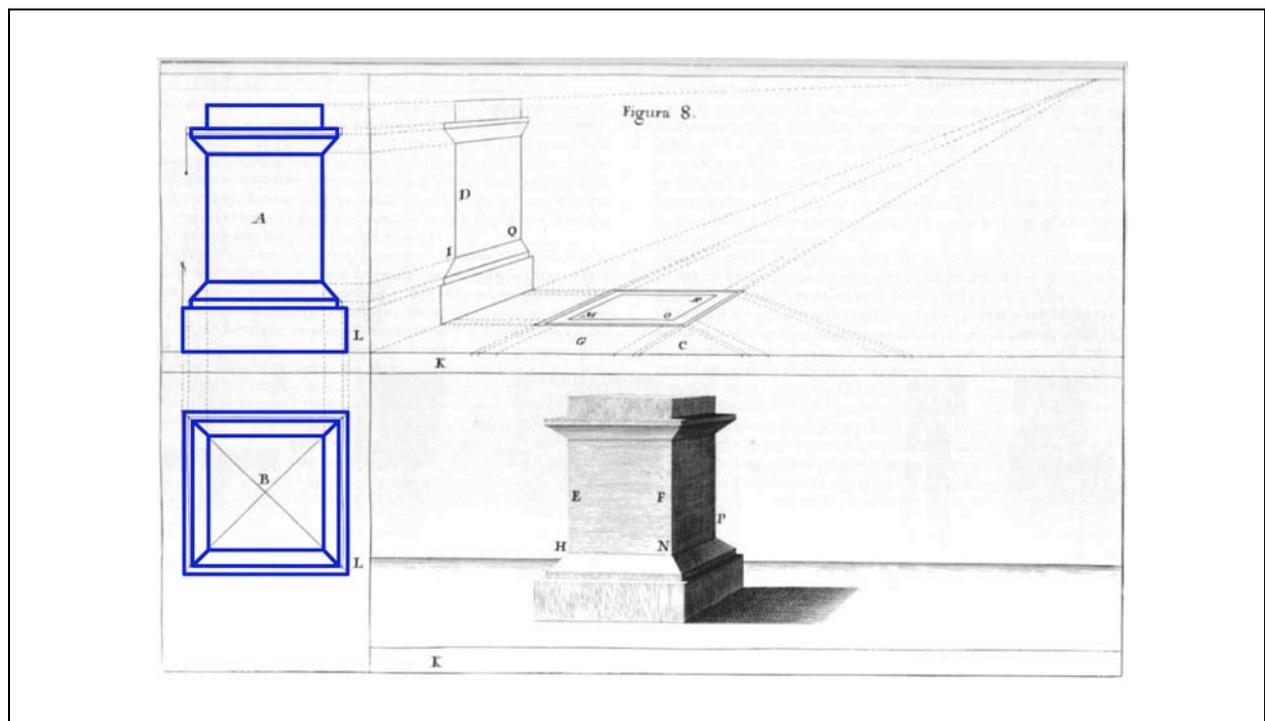
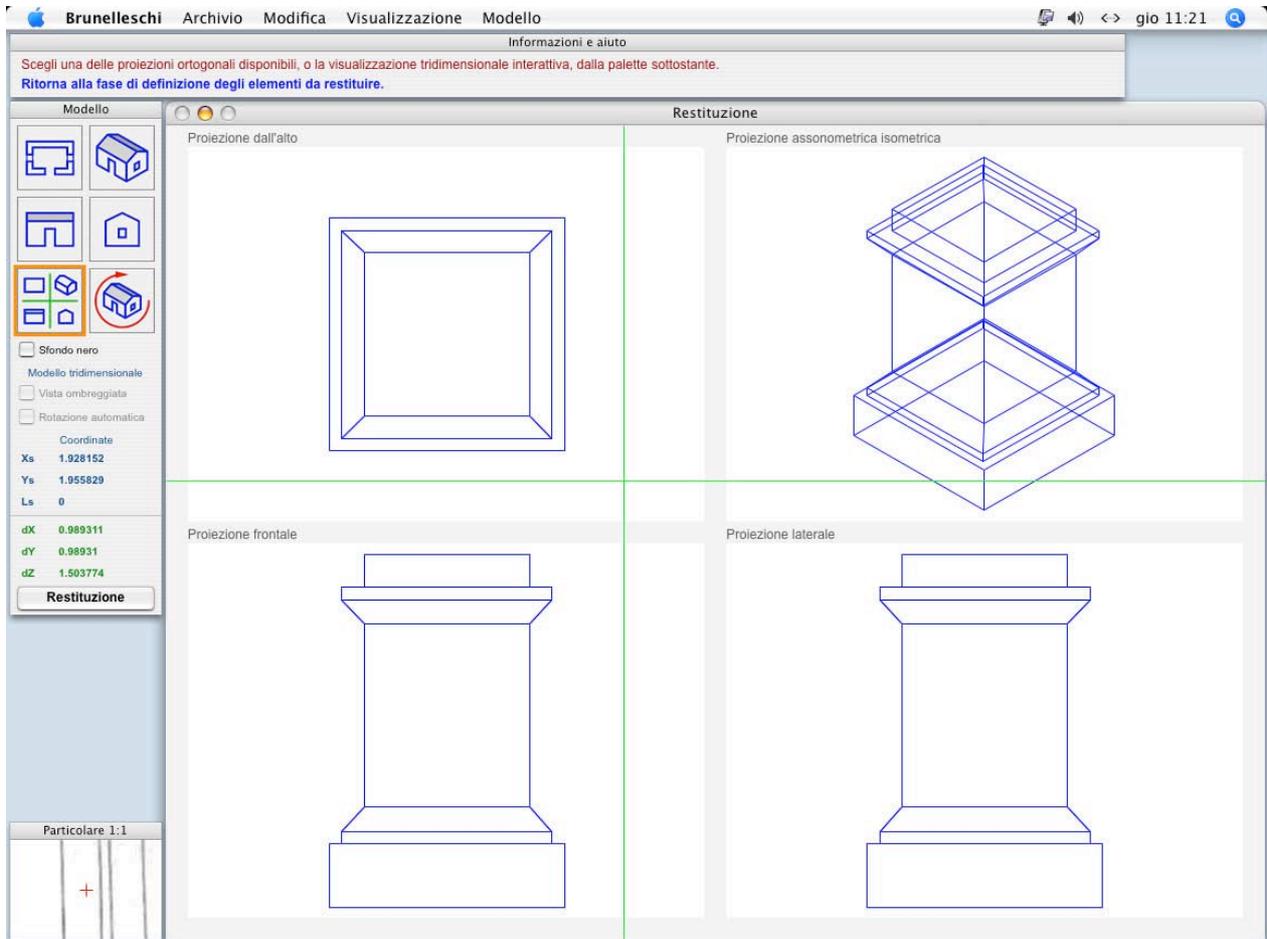
L'individuazione dei volumi mancanti viene completata aggiungendo le cornici, il piedistallo centrale e il dado sommitale.

Il modello tridimensionale viene aggiornato, in tempo reale, man mano che si procede all'individuazione degli elementi da restituire<sup>74</sup>; il software *Brunelleschi* consente di verificare in ogni momento il risultato del processo di restituzione, come mostrato in figura in alto nella pagina successiva.

La sovrapposizione delle proiezioni restituite alla figura ottava confermano l'esattezza e il grado di accuratezza ottenuto dall'operazione di restituzione.

Il modello tridimensionale restituito può essere esportato verso altri *software* per successive elaborazioni grafiche, come mostrato in figura nella pagina successiva.

<sup>74</sup> Visualizzazione del modello restituito a pagg.78-80

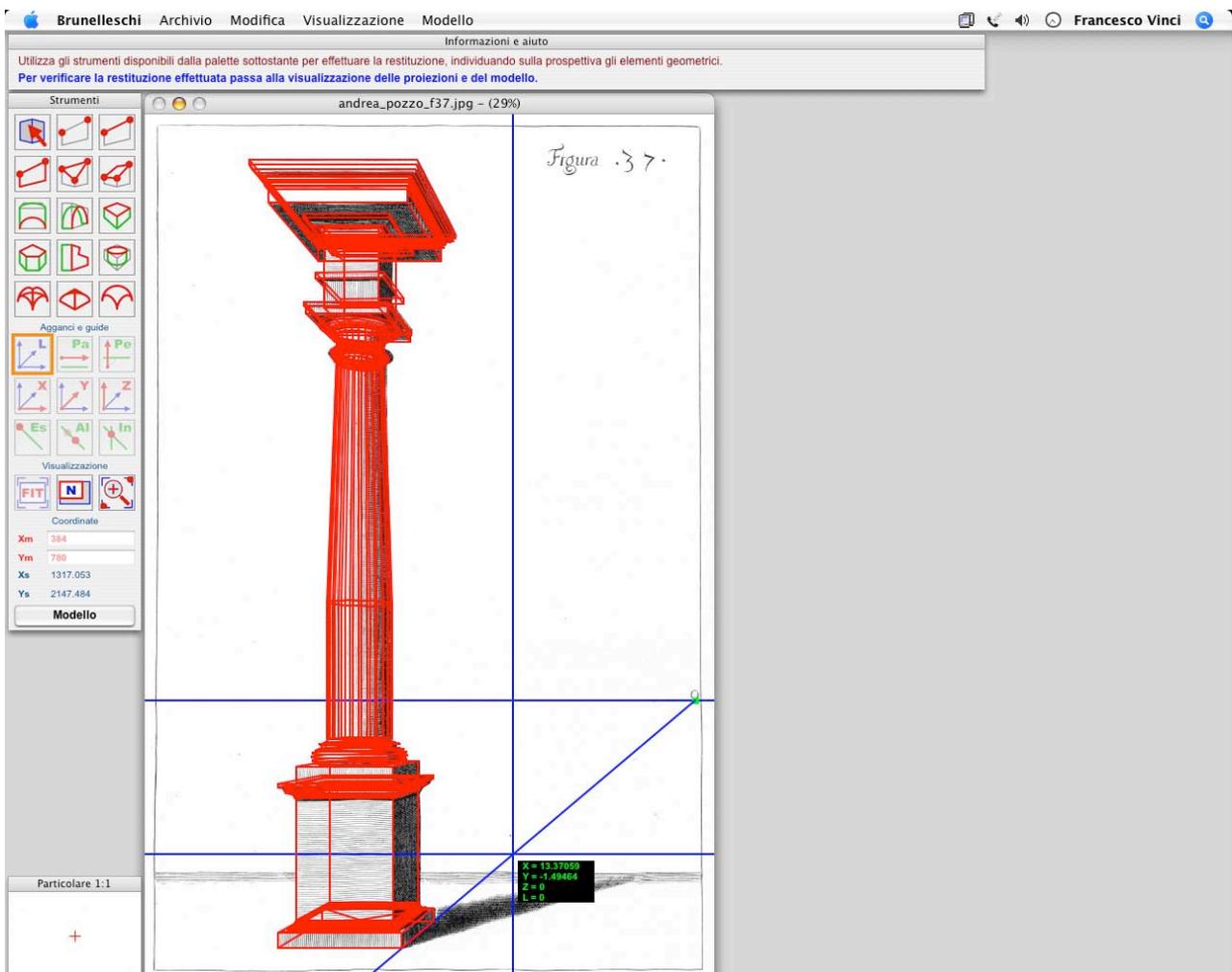


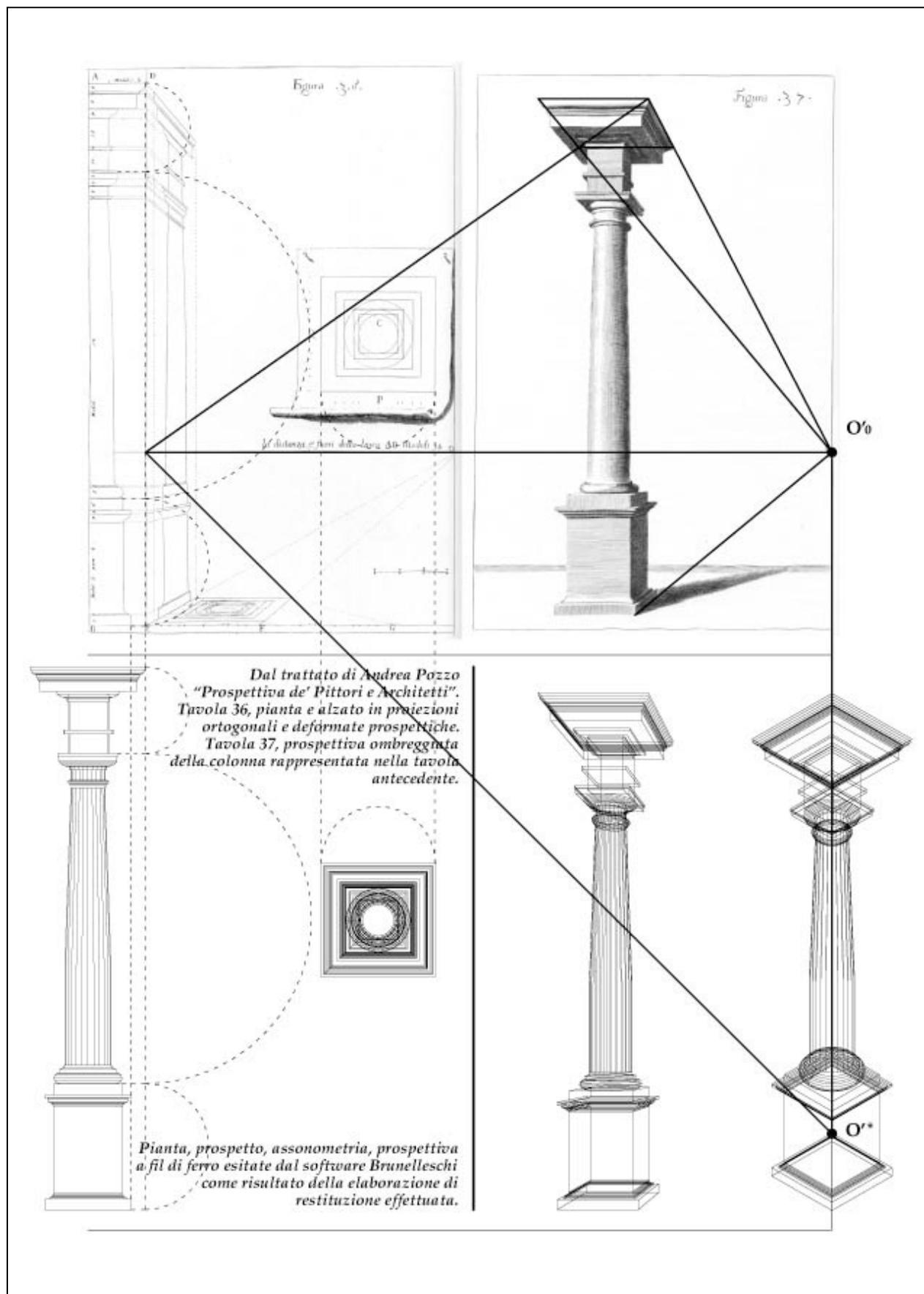
## RESTITUZIONE PROSPETTICA ASSISTITA

*Una colonna toscana rappresentata in prospettiva in figura 37 del trattato "Prospettiva de' Pittori e Architetti" di Andrea Pozzo.*

Come dato di partenza viene assunto il quadrato orientato secondo gli assi cartesiani principali che rappresenta nella prospettiva il profilo superiore della cornice della trabeazione.

Gli elementi geometrici della colonna vengono individuati dettagliatamente utilizzando gli strumenti di definizione delle forme disponibili nel software *Brunelleschi*.



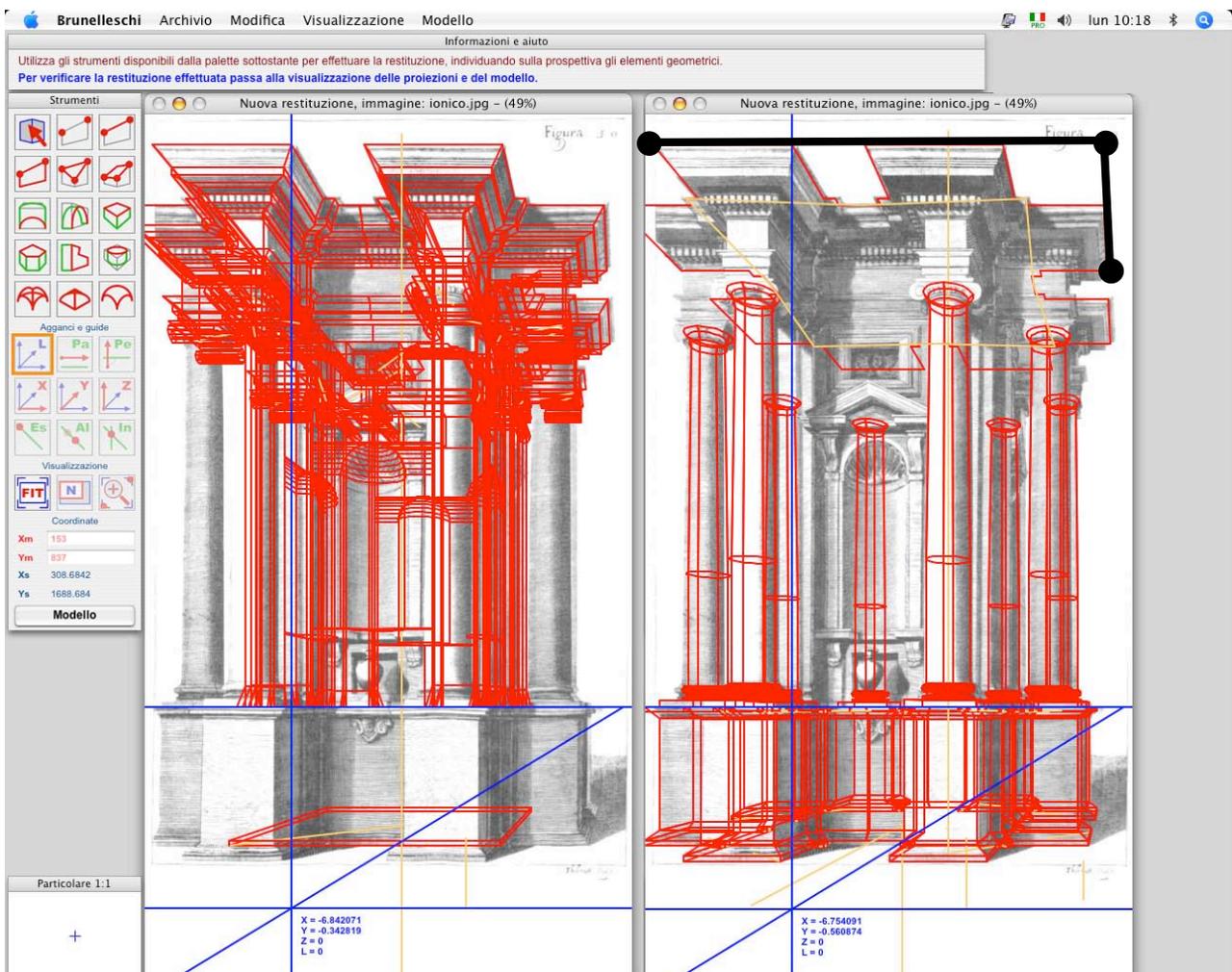


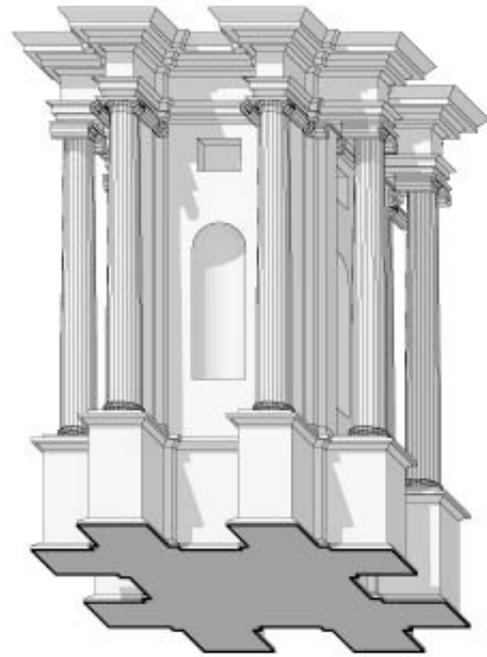
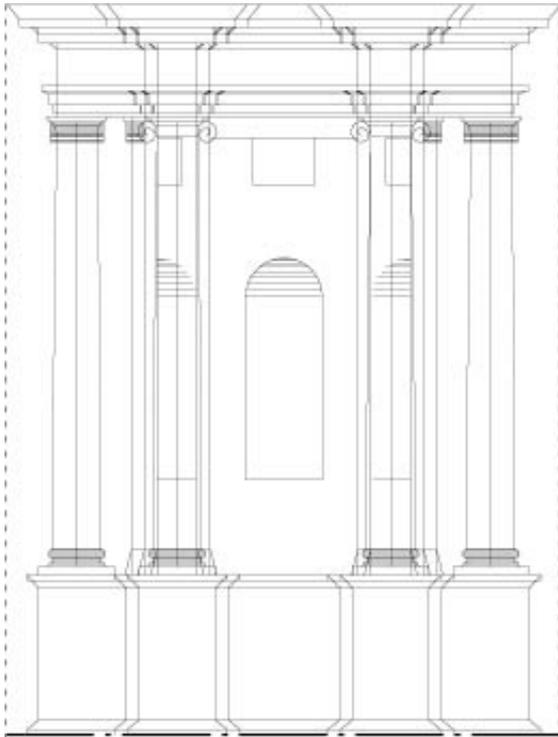
## RESTITUZIONE PROSPETTICA ASSISTITA

*Un edificio in stile ionico rappresentato in prospettiva in figura 50 del trattato "Prospettiva de' Pittori e Architetti" di Andrea Pozzo.*

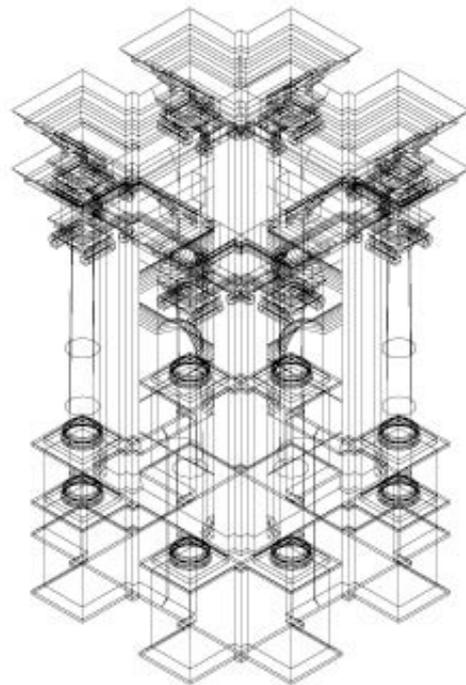
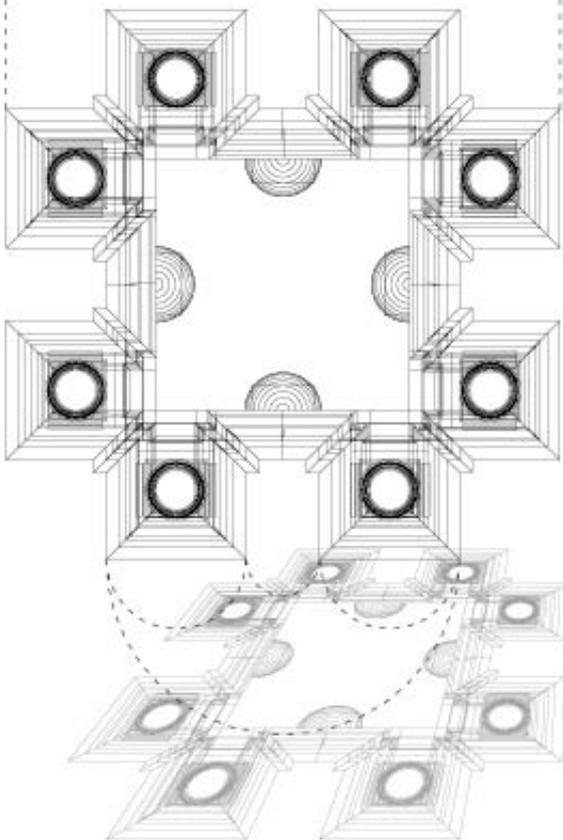
Come dato di partenza viene assunto il rettangolo orientato secondo gli assi cartesiani principali che rappresenta nella prospettiva il profilo superiore della cornice della trabeazione, come mostrato in figura sottostante.

Gli elementi geometrici dell'edificio vengono individuati utilizzando gli strumenti di definizione delle forme disponibili nel software *Brunelleschi*.





*Pianta, prospetto, assonometrie,  
prospettiva a fil di ferro esitate  
dal software Brunelleschi come  
risultato della elaborazione  
di restituzione effettuata.*

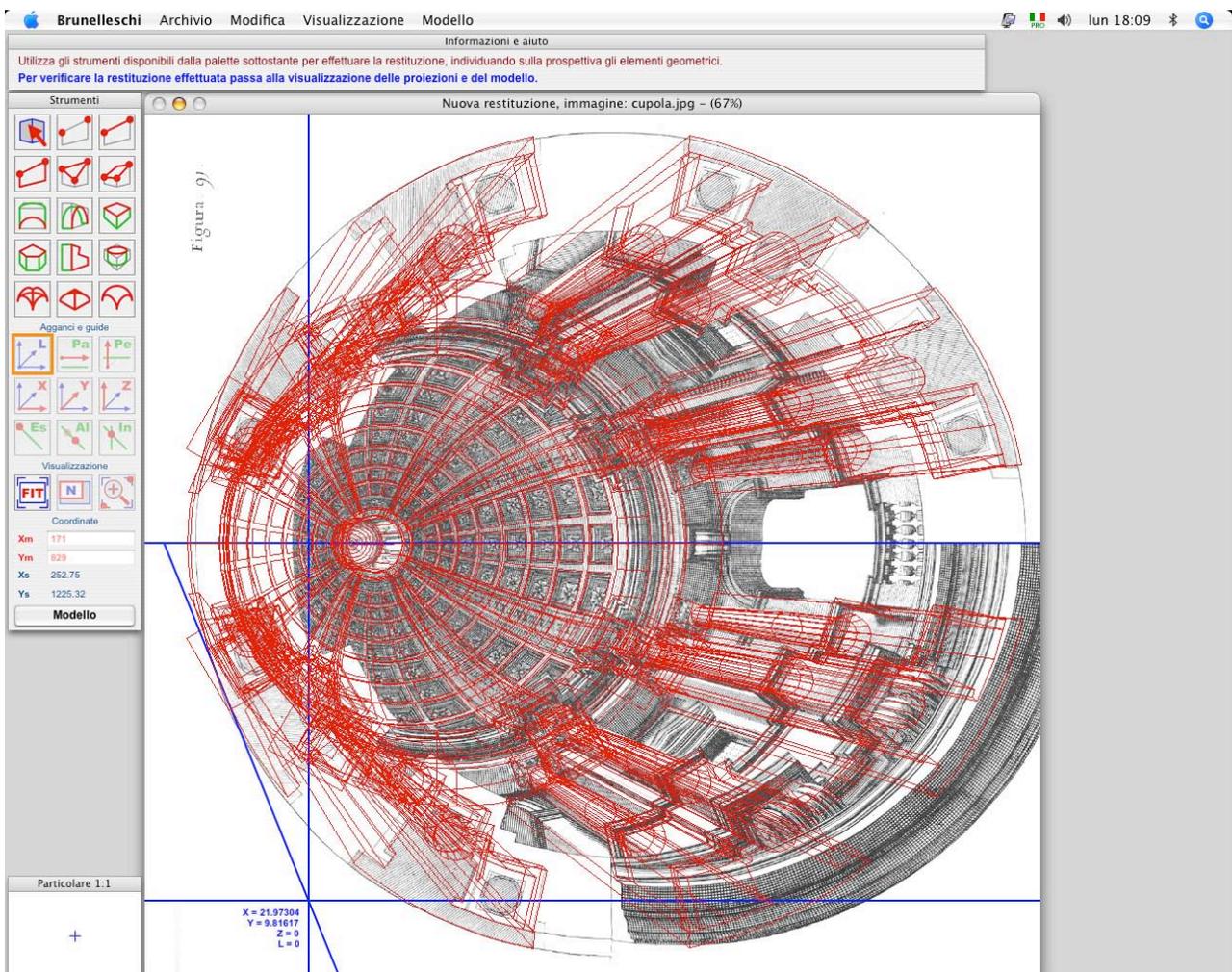


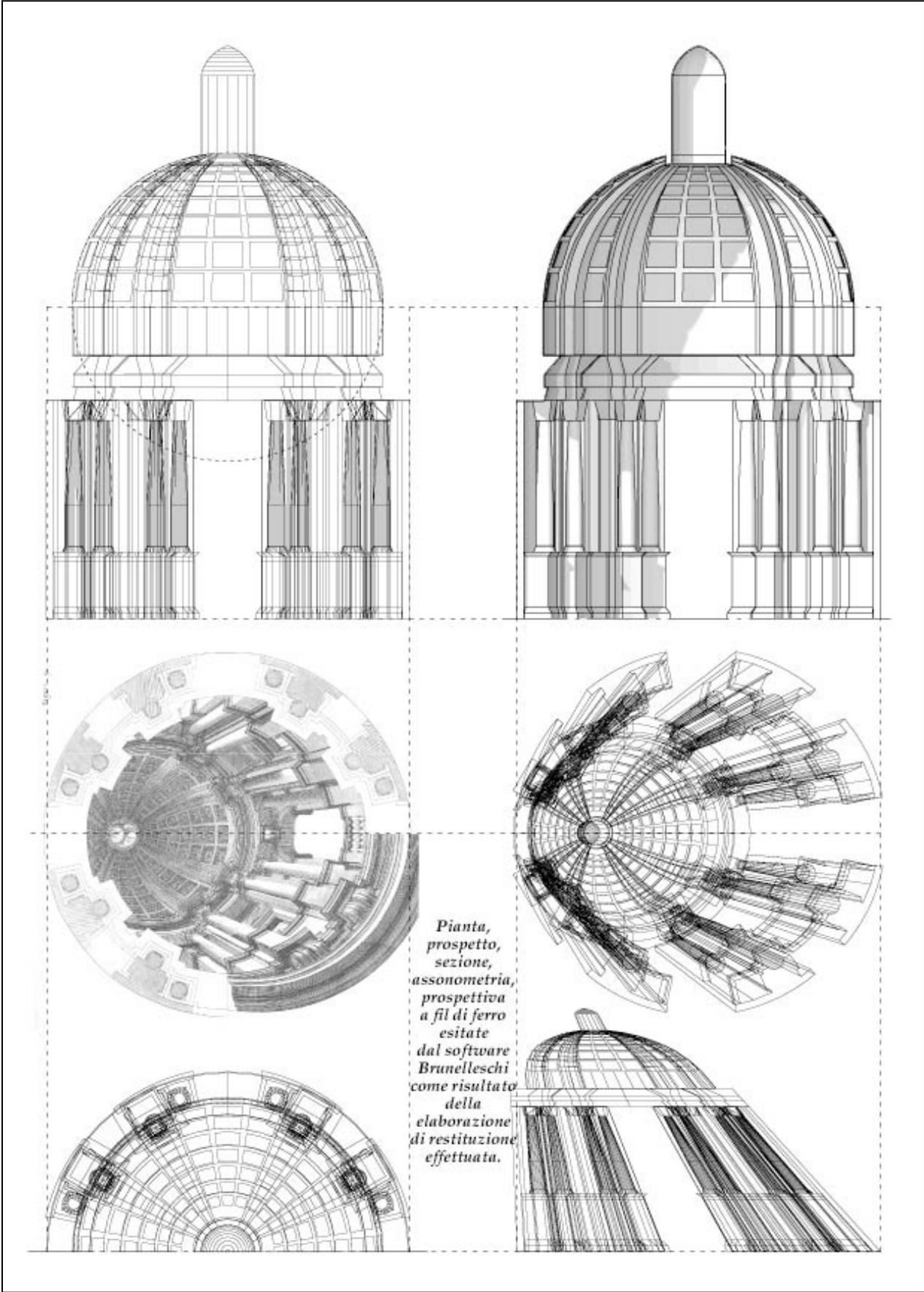
## RESTITUZIONE PROSPETTICA ASSISTITA

*Una cupola rappresentata in prospettiva dal basso in figura 91 del trattato "Prospettiva de' Pittori e Architetti" di Andrea Pozzo.*

Come dato di partenza viene assunto un rettangolo verticale rappresentato nella figura 90 del trattato, preparatoria della presente, avente come dimensioni il raggio della circonferenza e l'altezza della parte basamentale della cupola.

Gli elementi geometrici della cupola vengono individuati utilizzando gli strumenti di definizione delle forme disponibili nel software *Brunelleschi*.





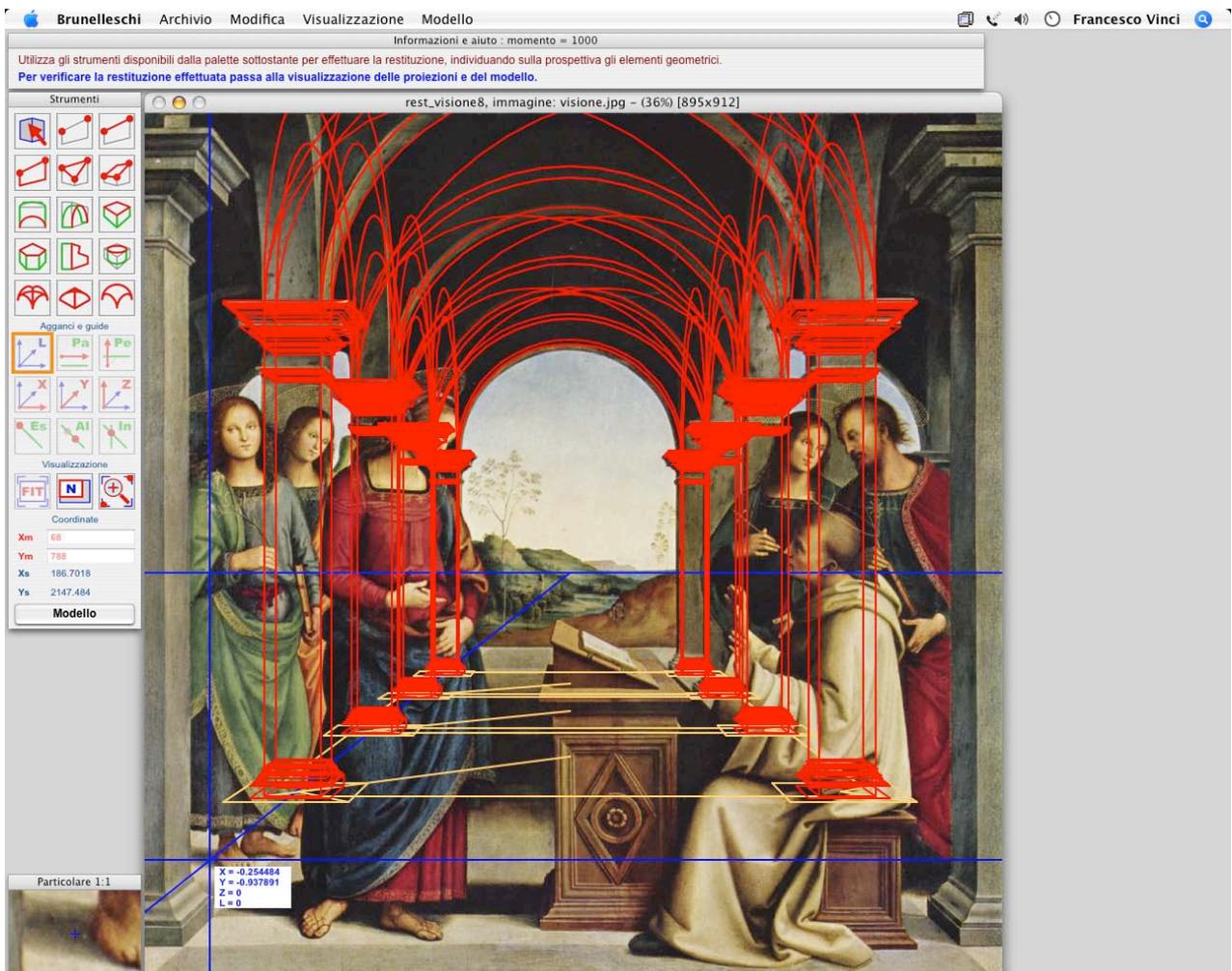
*Pianta,  
prospetto,  
sezione,  
assonometria,  
prospettiva  
a fil di ferro  
esitate  
dal software  
Brunelleschi  
come risultato  
della  
elaborazione  
di restituzione  
effettuata.*

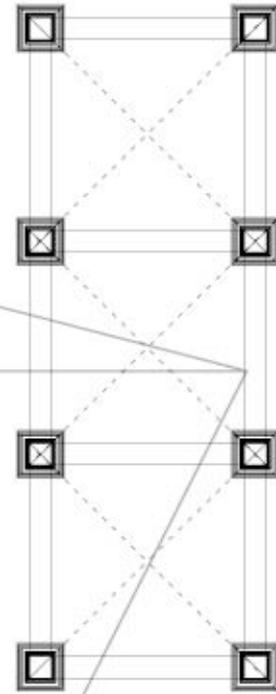
## RESTITUZIONE PROSPETTICA ASSISTITA

*Il portico voltato rappresentato in prospettiva nel dipinto "L'apparizione della Vergine a San Bernardo" (1493) di Pietro Perugino.*

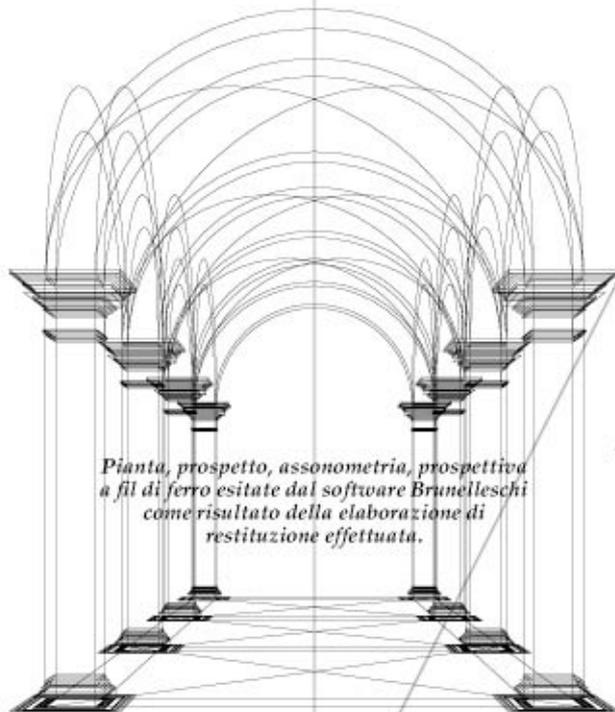
Come dato di partenza viene assunto il quadrato d'imposta della cellula voltata, confermata dalla corrispondenza della forma quadrata dei pilastri.

Gli elementi geometrici del portico vengono individuati dettagliatamente utilizzando gli strumenti di definizione delle forme disponibili nel software *Brunelleschi*.

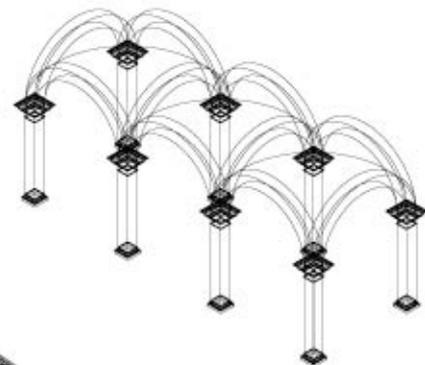
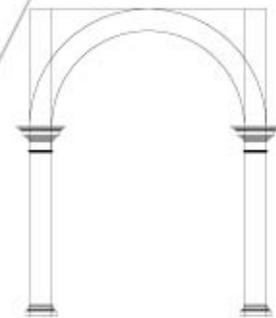




*Come ipotesi di partenza viene assunta la base quadrata della cellula voltata, confermata dal software verificando la forma quadrata delle basi dei pilastri*



*Pianta, prospetto, assonometria, prospettiva a fil di ferro esitate dal software Brunelleschi come risultato della elaborazione di restituzione effettuata.*



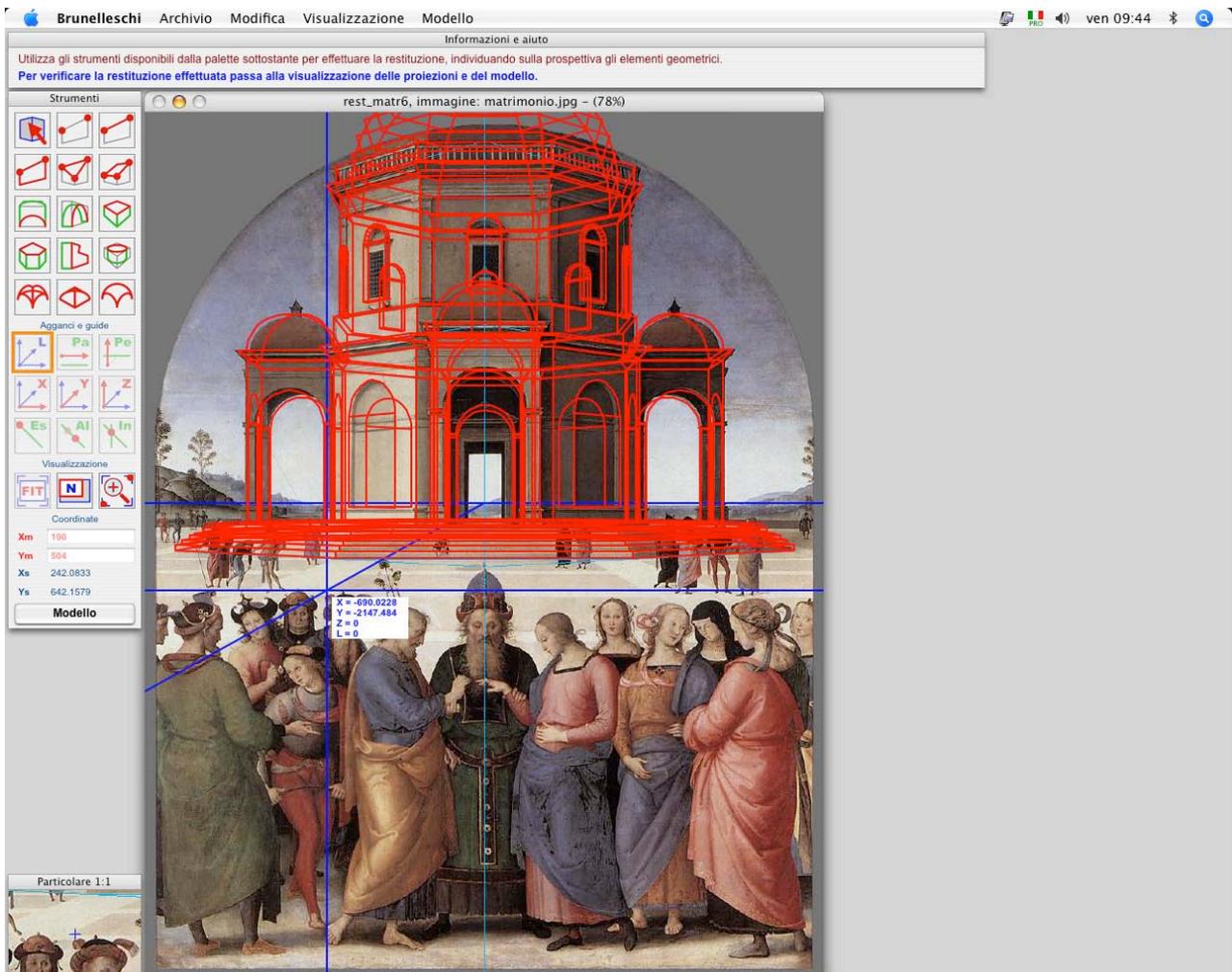
O<sup>ra</sup>

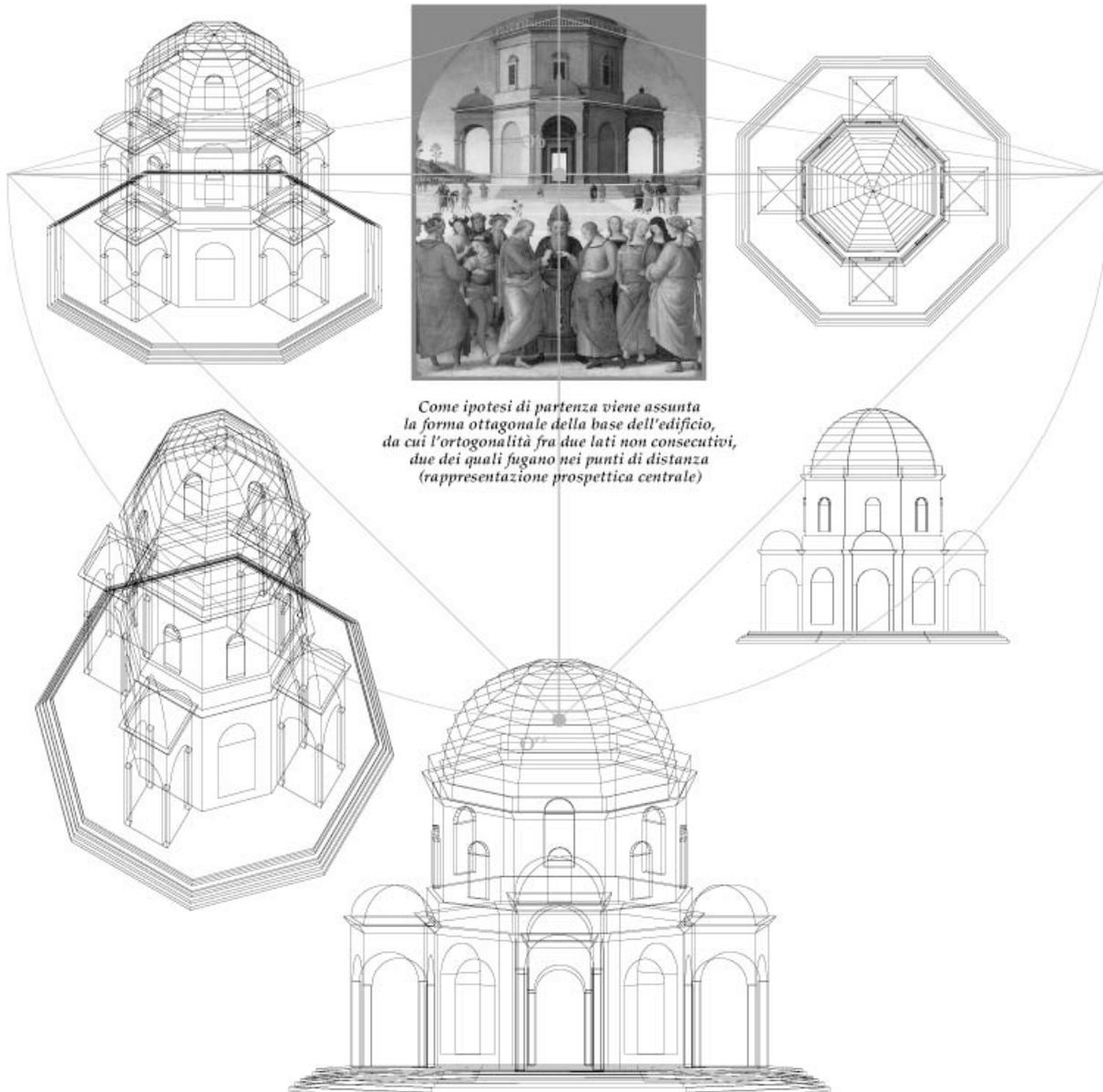
## RESTITUZIONE PROSPETTICA ASSISTITA

*Schema geometrico dell'edificio rappresentato in prospettiva nel dipinto "Il matrimonio della Vergine" (1504) di Pietro Perugino.*

Come dato di partenza viene assunta la forma ottagonale della base dell'edificio, e l'angolo retto formato da due lati non contigui dell'ottagono.

Gli elementi geometrici dell'edificio vengono individuati schematicamente utilizzando gli strumenti di definizione delle forme disponibili nel software *Brunelleschi*.





*Come ipotesi di partenza viene assunta la forma ottagonale della base dell'edificio, da cui l'ortogonalità fra due lati non consecutivi, due dei quali fuggano nei punti di distanza (rappresentazione prospettica centrale)*

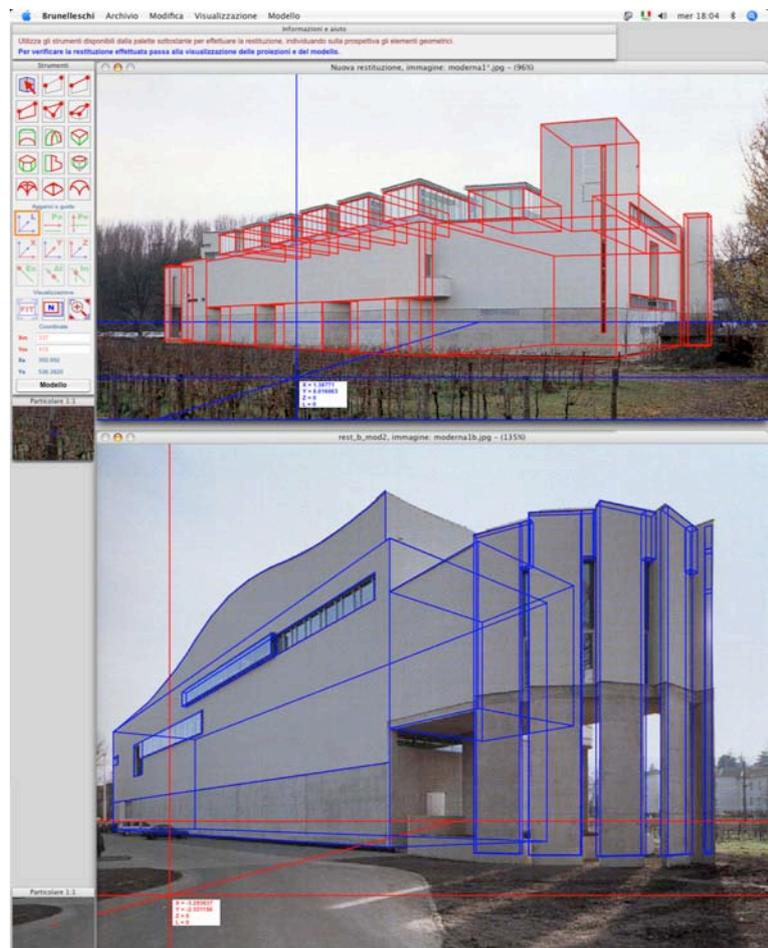
*Pianta, prospetto, assonometrie, prospettiva a fil di ferro esitate dal software Brunelleschi come risultato della elaborazione di restituzione effettuata.*

## RESTITUZIONE PROSPETTICA ASSISTITA

*IL museo per la collezione Essl edificato a Klosterneuburg bei Wien, progetto redatto da Heinz Tesar (1999).*

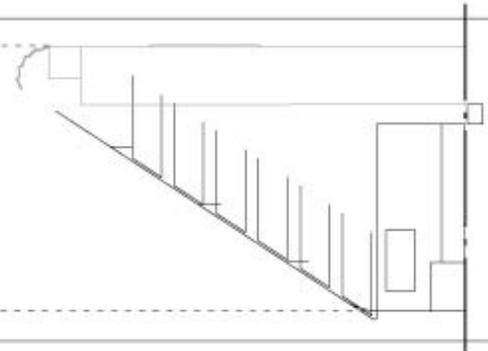
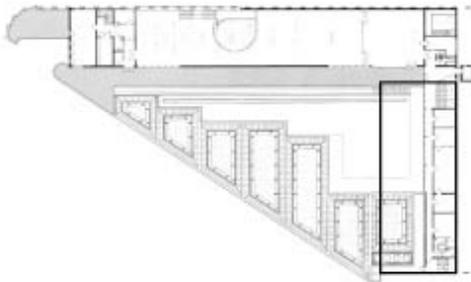
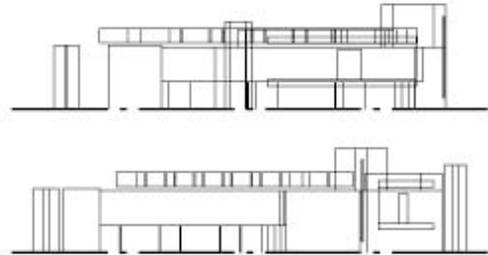
Come dati di partenza vengono assunti i rettangoli, rispettivamente orizzontale e verticale, indicati in figura nella pagina a fianco.

Gli elementi geometrici del museo vengono individuati schematicamente utilizzando gli strumenti di definizione delle forme disponibili nel *software Brunelleschi*.

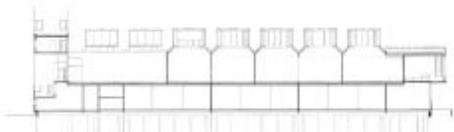
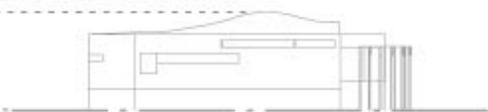
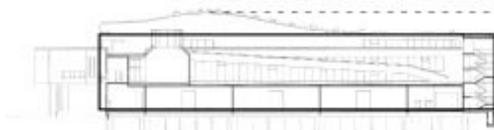
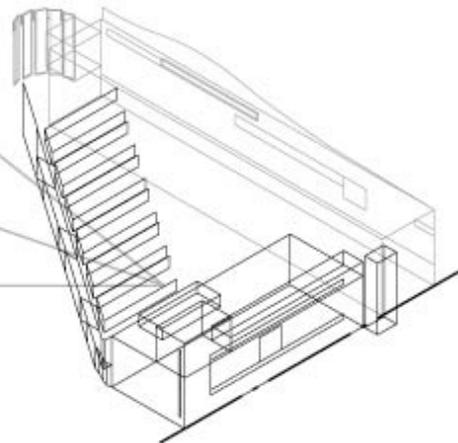




*Come dati di partenza vengono assunte due misure in pianta appartenenti a rette fra di loro ortogonali*



*Come dati di partenza vengono assunte due misure in alzato giacenti su un piano perpendicolare al geometrico e appartenenti a rette fra di loro ortogonali*



*Piante, prospetti, assonometrica a fil di ferro esitate dal software Brunelleschi come risultato della elaborazione di restituzione effettuata.*

*RIPRODUZIONE DELLE TAVOLE GRAFICHE*



**TAVOLA 1**

Formato 50x35 centimetri

Intestazione:

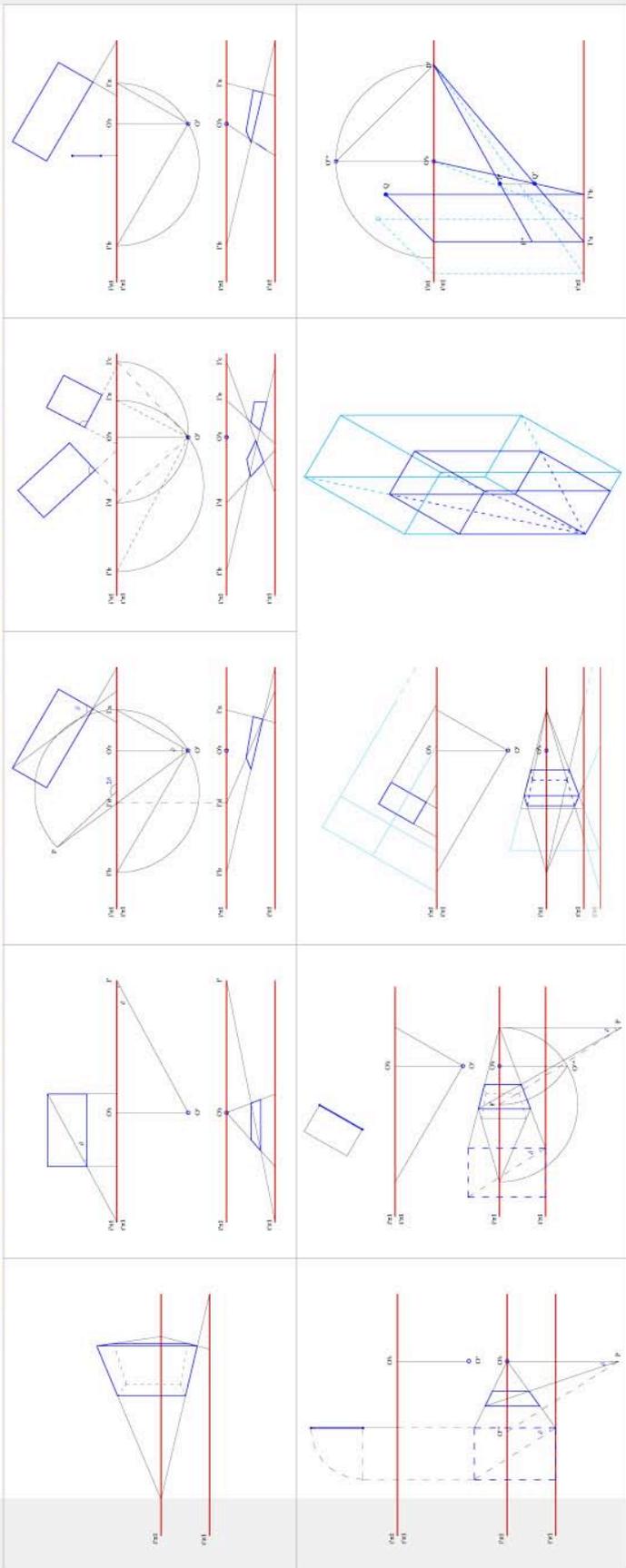
**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

Titolo:

**FONDAMENTI SCIENTIFICI E**  
**PRESENTAZIONE DEL SOFTWARE**

Procedimenti geometrici di base codificati nel software Brunelleschi per la determinazione dell'orientamento di una rappresentazione prospettica e per la susseguente restituzione in vera forma



**Importazione della rappresentazione prospettica.** consistente nelle operazioni atte ad ottenere l'immagine prospettica di cui si vuole eseguire la restituzione in formato digitale di tipo raster, eventualmente corredata di opportuni elementi geometrici di riferimento per facilitare la fase di restituzione.

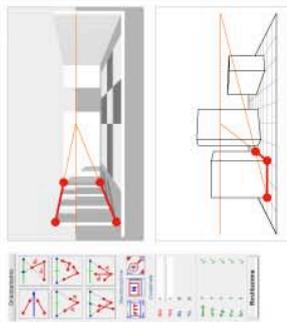
**1 Determinazione dell'orientamento della prospettiva.** consistente nell'individuazione, con un'operazione di "riciclo" sull'immagine, degli elementi geometrici che consentono la determinazione dell'orizzonte, del punto di vista, e della distanza principale, e con essi lo "spazio tridimensionale" della prospettiva.

**2 Definizione degli elementi geometrici della prospettiva.** consistente nell'individuazione, con un'operazione "ritocco" sulla restituzione, dei punti di vista, del punto di vista, del punto di vista, e del punto di vista, e con essi lo "spazio tridimensionale" della prospettiva.

**3 Gestione del modello restituito.** consistente nella visualizzazione, controllo e verifica, del modello tridimensionale restituito, nelle sue proiezioni ortogonali e assonometriche, nella sua "realtà" spaziale, compresa l'esportazione del modello o degli elaborati bidimensionali verso altri software per successive elaborazioni.

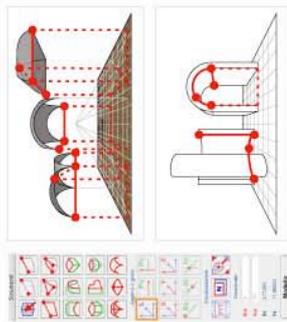
### 1

**Determinazione dell'orientamento della prospettiva**



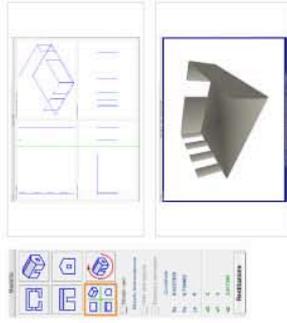
### 2

**Definizione degli elementi geometrici della prospettiva**



### 3

**Gestione del modello restituito**



UNIVERSITA' DEGLI STUDI DI CATANIA • FACOLTA' DI ARCHITETTURA • Dottorato di Ricerca in TEORIA E STORIA DELLA RAPPRESENTAZIONE, XIX CICLO  
 BRUNELLESCHI, software per la restituzione prospettica assistita • Tesi di Dottorato di Francesco Vinci • Coordinatore e Tutor, Prof. Arch. Giuseppe Pagnano  
 FONDAMENTI SCIENTIFICI E PRESENTAZIONE DEL SOFTWARE

**TAVOLA 2**

Formato 50x35 centimetri

Intestazione:

**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

Titolo:

**PRESENTAZIONE DEL SOFTWARE**



Superficie di rivoluzione



**TAVOLA 3**

Formato 50x35 centimetri

Intestazione:

**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

Titolo:

**PRESENTAZIONE DEL SOFTWARE**



**TAVOLA 4**

Formato 50x35 centimetri

Intestazione:

**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

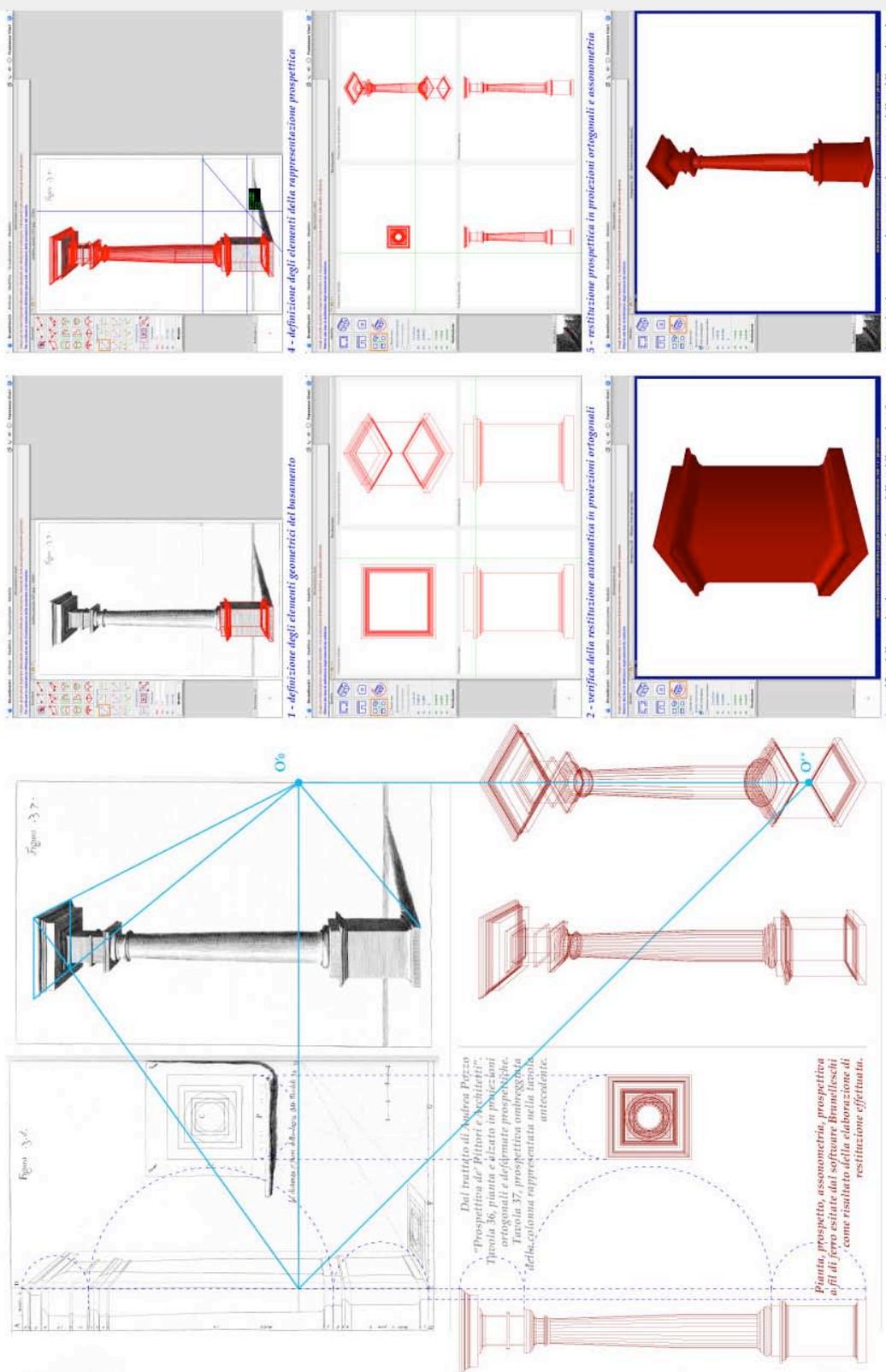
Titolo:

**RESTITUZIONE PROSPETTICA ASSISTITA**

Oggetto:

*Una colonna toscana rappresentata in prospettiva in*  
*figura 37 del trattato "Prospettiva de' Pittori e*  
*Architetti" di Andrea Pozzo.*

Restituzione prospettica assistita con l'utilizzo del software *Brunelleschi*, una colonna toscana rappresentata in prospettiva nella tavola 37 del trattato "Prospettiva de' Pittori e Architetti" di Andrea Pozzo



Dal trattato di Andrea Pozzo "Prospettiva de' Pittori e Architetti", Tavola 36, pianta di Alicano in posizione assonale, e Tavola 37, prospettiva completa della colonna rappresentata nella tavola antecedente.

Pianta, prospettiva, assonometria, prospettiva di un ferro estive dal software Brunelleschi, come risultato della ridisegnazione di restituzione effettuate.



**TAVOLA 5**

Formato 50x35 centimetri

Intestazione:

**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

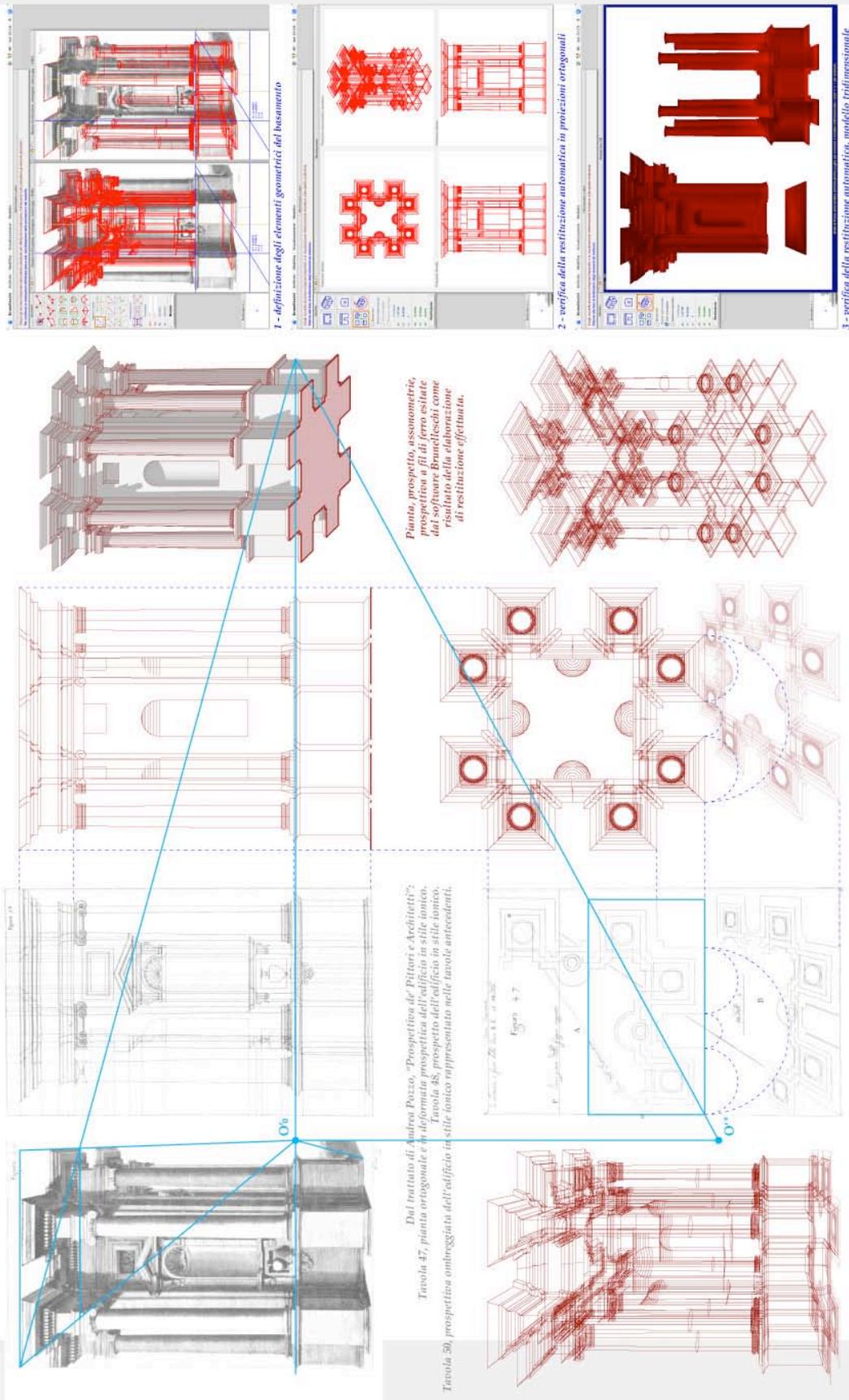
Titolo:

**RESTITUZIONE PROSPETTICA ASSISTITA**

Oggetto:

*Un edificio in stile ionico rappresentato in prospettiva in  
figura 50 del trattato "Prospettiva de' Pittori e  
Architetti" di Andrea Pozzo.*

Restituzione prospettica assistita con l'utilizzo del software Brunelleschi, una edificio in stile ionico rappresentato in prospettiva nella tavola 50 del trattato "Prospettiva de' Pittori e Architetti" di Andrea Pozzo



UNIVERSITA' DEGLI STUDI DI CATANIA • FACOLTA' DI ARCHITETTURA • Dottorato di Ricerca in TEORIA E STORIA DELLA RAPPRESENTAZIONE, XIX CICLO  
BRUNELLESCHI, software per la restituzione prospettica assistita • Tesi di Dottorato di Francesco Vinci • Coordinatore e Tutor, Prof. Arch. Giuseppe Pagnano  
RESTITUZIONE PROSPETTICA ASSISTITA



**TAVOLA 6**

Formato 50x35 centimetri

Intestazione:

**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

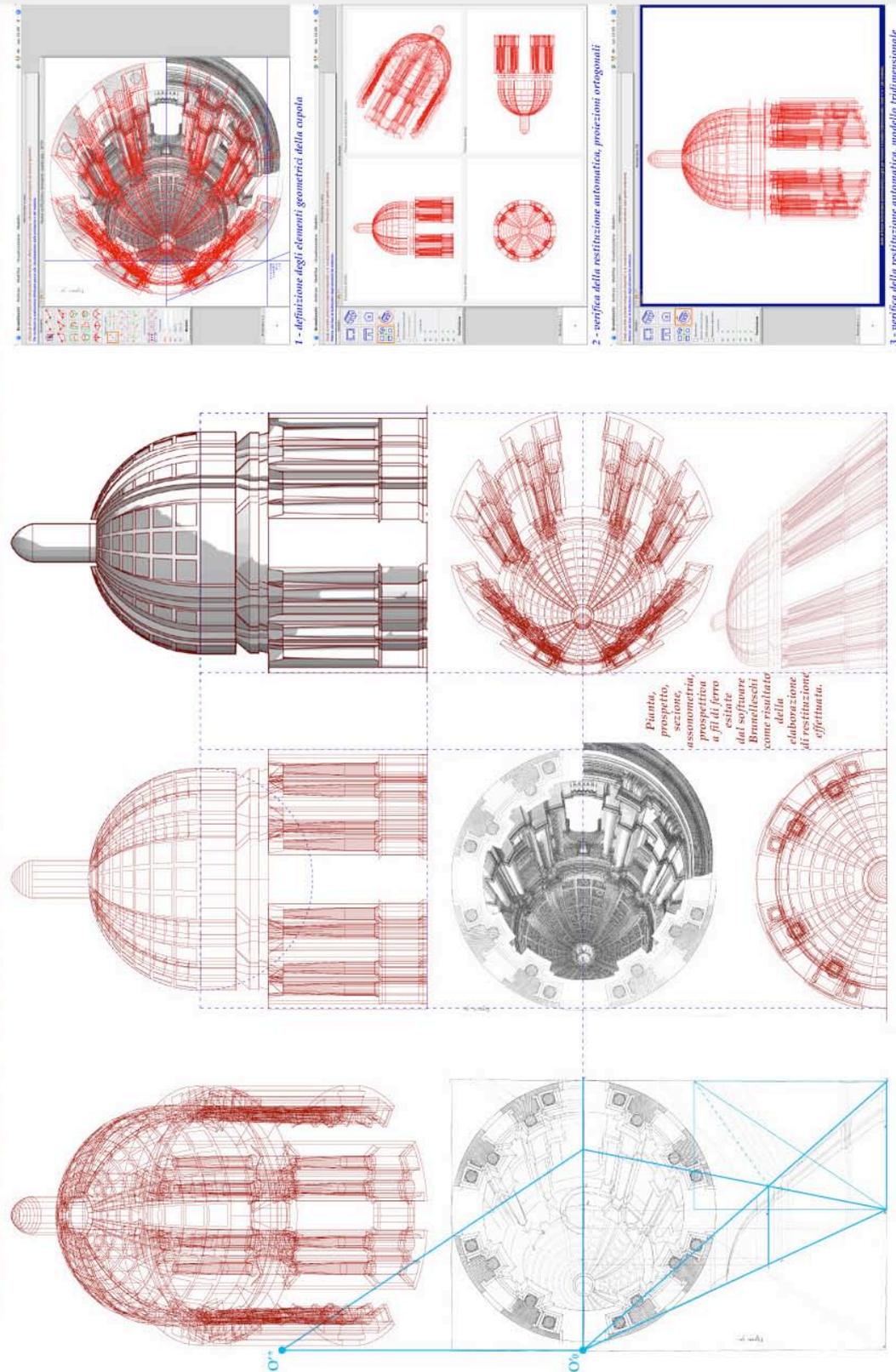
Titolo:

**RESTITUZIONE PROSPETTICA ASSISTITA**

Oggetto:

*Una cupola rappresentata in prospettiva dal basso in*  
*figura 91 del trattato "Prospettiva de' Pittori e*  
*Architetti" di Andrea Pozzo.*

Restituzione prospettica assistita con l'utilizzo del software Brunelleschi, una cupola rappresentata in prospettiva dal basso nella tavola 91 del trattato "Prospettiva de' Pittori e Architetti" di Andrea Pozzo



UNIVERSITA' DEGLI STUDI DI CATANIA • FACOLTA' DI ARCHITETTURA • Dottorato di Ricerca in TEORIA E STORIA DELLA RAPPRESENTAZIONE, XIX CICLO  
 BRUNELLESCHI, software per la restituzione prospettica assistita • Tesi di Dottorato di Francesco Vinci • Coordinatore e Tutor, Prof. Arch. Giuseppe Pagliano  
 RESTITUZIONE PROSPETTICA ASSISTITA



**TAVOLA 7**

Formato 50x35 centimetri

Intestazione:

**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

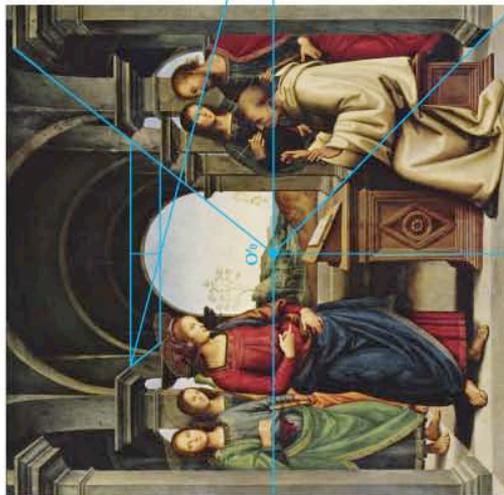
Titolo:

**RESTITUZIONE PROSPETTICA ASSISTITA**

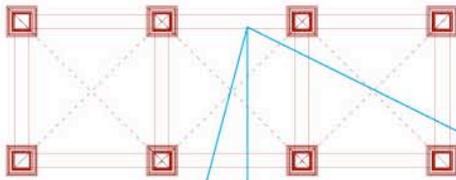
Oggetto:

*Il portico voltato rappresentato in prospettiva nel  
dipinto "L'apparizione della Vergine a San Bernardo"  
(1493) di Pietro Perugino.*

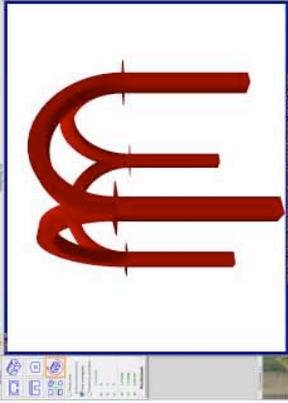
Restituzione prospettica assistita con l'utilizzo del software *Brunelleschi*, il portico voltato rappresentato in prospettiva nel dipinto "L'apparizione della Vergine a San Bernardo" (1493) di Pietro Perugino



Come ipotesi di partenza viene assunta la base quadrata della cellula voltata, confermata dal software verificando la forma quadrata delle basi dei pilastri



1 - definizione degli elementi geometrici strutturali di base



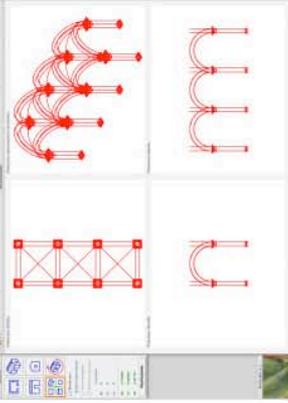
2 - verifica della restituzione automatica, modello tridimensionale



3 - possibile errore prospettico, data la forma apparente dello scrittoio



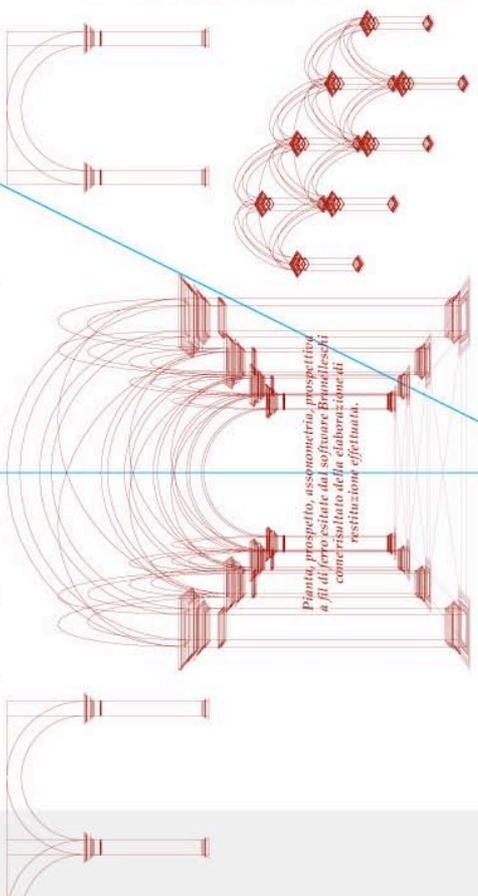
4 - definizione degli elementi della rappresentazione prospettica



5 - restituzione prospettica in proiezioni ortogonali e assonometria



6 - restituzione prospettica automatica come modello tridimensionale



Pianta, prospetto, assonometria, prospettiva  
6 fili di ferro esitate dal software Brunelleschi  
come risultato della elaborazione di  
restituzione effettiva.



**TAVOLA 8**

Formato 50x35 centimetri

Intestazione:

**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

Titolo:

**RESTITUZIONE PROSPETTICA ASSISTITA**

Oggetto:

*Schema geometrico dell'edificio rappresentato in*  
*prospettiva nel dipinto "Il matrimonio della Vergine"*  
*(1504) di Pietro Perugino.*

1 - definizione degli elementi geometrici strutturali di base

2 - verifica della restituzione automatica, modello tridimensionale

3 - possibile errore prospettico, data la forma apparente dello scrittoio

4 - definizione degli elementi della rappresentazione prospettica

5 - restituzione prospettica in proiezioni ortogonali e assonometria

6 - restituzione prospettica automatica come modello tridimensionale

Come ipotesi di partenza viene assunta la forma ottagonale della base dell'edificio, da cui l'ortogonalità. Tra due lati non consecutivi, due dei quali fungono nei punti di distanza (rappresentazione prospettica centrale)

Pianta, prospetto, assonometria, prospettiva in fili di ferro esitate dal software Brunelleschi come risultato della elaborazione di restituzioni effettuate.



**TAVOLA 9**

Formato 50x35 centimetri

Intestazione:

**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

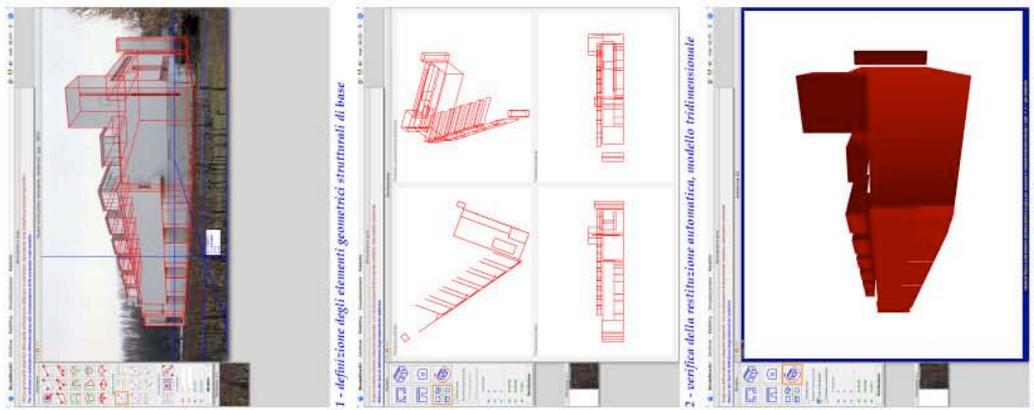
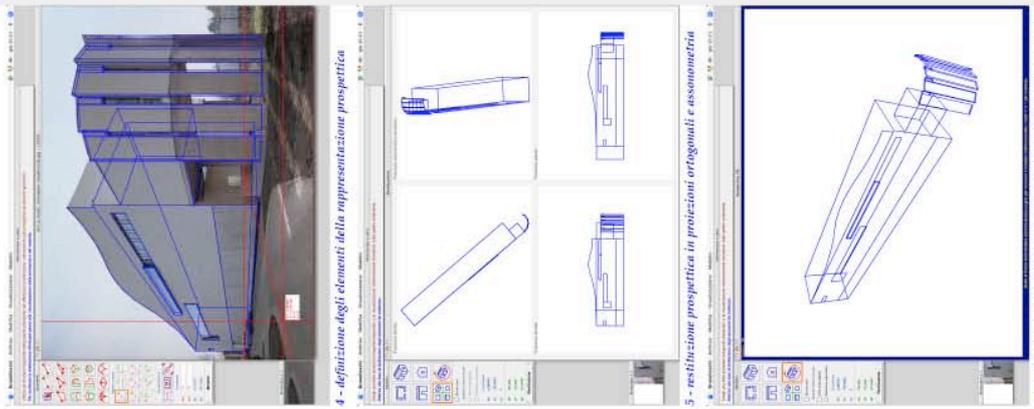
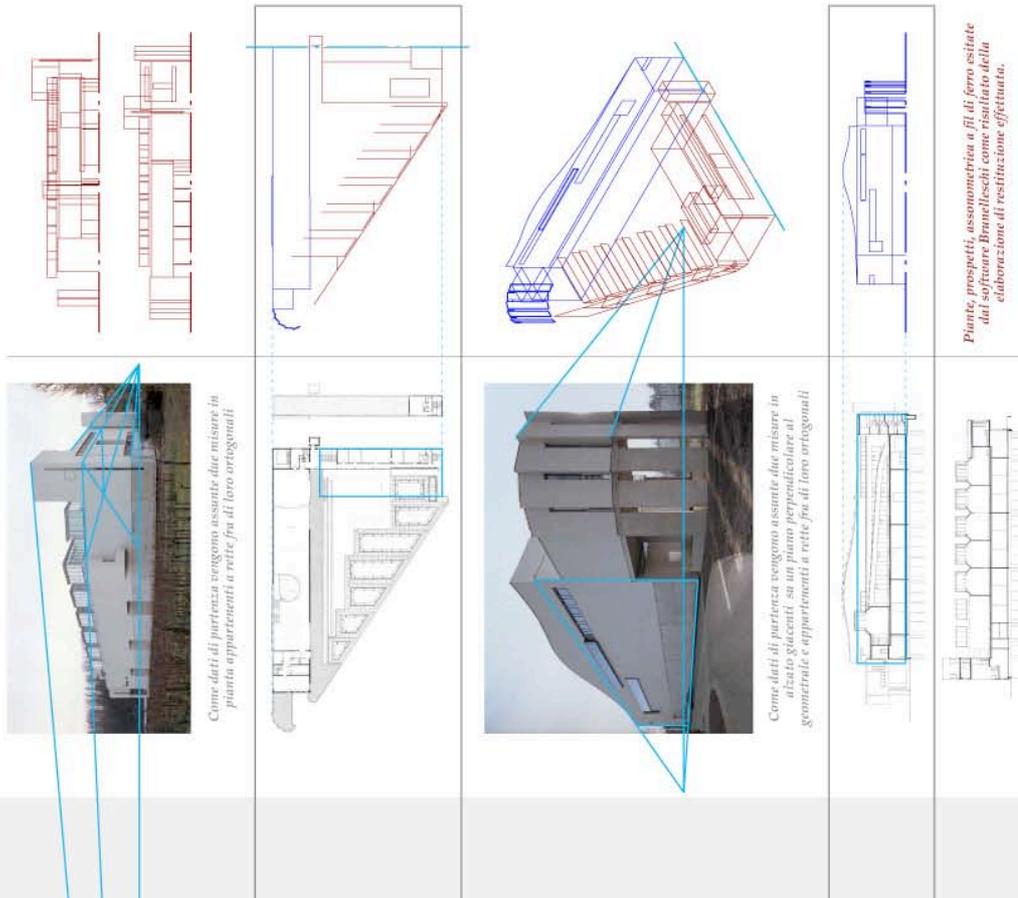
Titolo:

**RESTITUZIONE PROSPETTICA ASSISTITA**

Oggetto:

*IL museo per la collezione Essl edificato a*  
*Klosterneuburg bei Wien,*  
*progetto redatto da Heinz Tesar (1999).*

Restituzione prospettica assistita da immagini fotografiche con l'utilizzo del software Brunelleschi, il museo per la collezione Essl edificato a Klosterneuburg bei Wien, progetto redatto da Heinz Tesar (1999)



6 - restituzione prospettica automatica come modello tridimensionale

3 - possibile errore prospettico, data la forma apparente dello scrittoio

UNIVERSITA' DEGLI STUDI DI CATANIA • FACOLTA' DI ARCHITETTURA • Dottorato di Ricerca in TEORIA E STORIA DELLA RAPPRESENTAZIONE, XIX CICLO  
 BRUNELLESCHI, software per la restituzione prospettica assistita • Tesi di Dottorato di Francesco Vinci • Coordinatore e Tutor, Prof. Arch. Giuseppe Pagnano  
 RESTITUZIONE PROSPETTICA ASSISTITA



**TAVOLA 10**

Formato 50x35 centimetri

Intestazione:

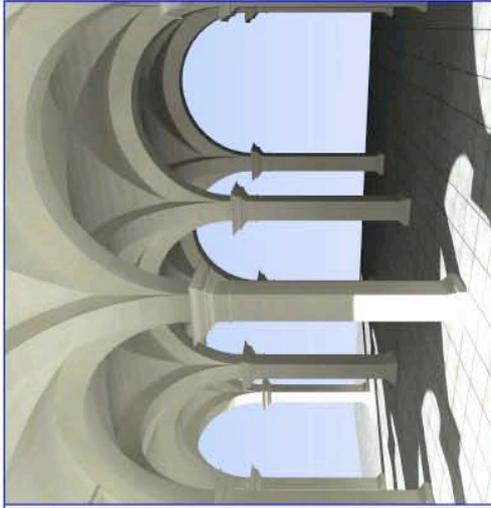
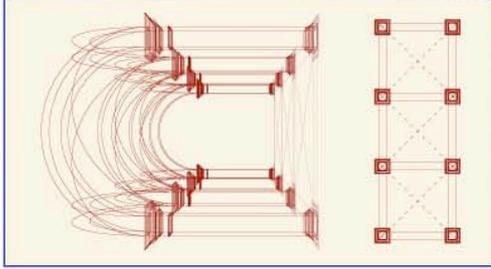
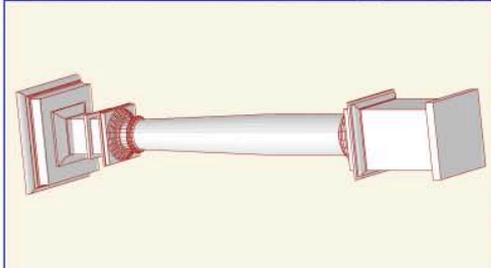
**UNIVERSITA' DEGLI STUDI DI CATANIA**  
**FACOLTÀ DI ARCHITETTURA**  
*Dottorato di Ricerca in*  
**TEORIA E STORIA DELLA RAPPRESENTAZIONE**  
**XIX CICLO**

**BRUNELLESCHI**  
*software per la restituzione prospettica assistita*  
*Tesi di Dottorato di*  
*Francesco Vinci*  
*Coordinatore e Tutor*  
*Prof. Arch. Giuseppe Pagnano*

Titolo:

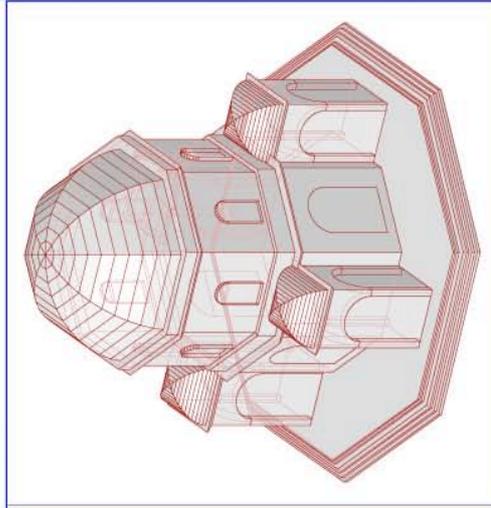
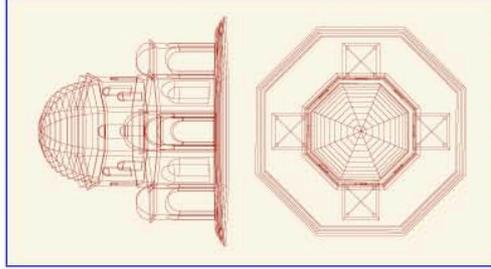
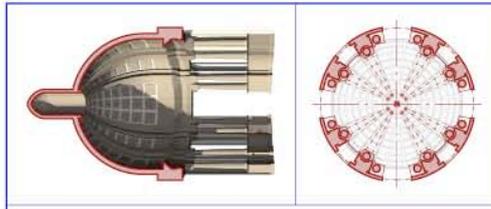
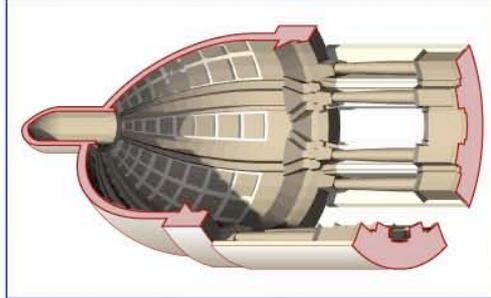
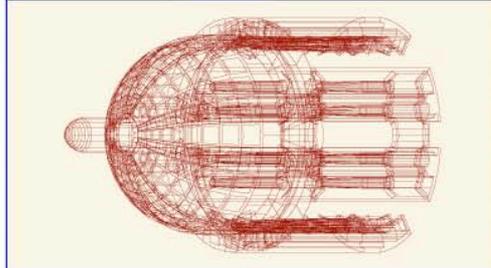
**RIELABORAZIONE DEI MODELLI**  
**TRIDIMENSIONALI RESTITUITI**

Rielaborazioni dei modelli tridimensionali esitati dal software Brunelleschi, render con ombre proprie e proiezioni ortogonali quotate dai modelli lineari...



Doc: Andrea Pozzo, fig.37, trattato "Prospettiva de' Pittori e Architetti"

Doc: Pietro Perugino, "L'apposizione nella Vergine e San Bernardo"



Doc: Andrea Pozzo, fig.91, trattato "Prospettiva de' Pittori e Architetti"

Doc: Pietro Perugino, "Il matrimonio della Vergine"



UNIVERSITA' DEGLI STUDI DI CATANIA • FACOLTA' DI ARCHITETTURA • Dottorato di Ricerca in TEORIA E STORIA DELLA RAPPRESENTAZIONE, XIX CICLO  
 BRUNELLESCHI, software per la restituzione prospettica assistita • Tesi di Dottorato di Francesco Vinci • Coordinatore e Tutor, Prof. Arch. Giuseppe Pagnano  
 RIELABORAZIONE DEI MODELLI TRIDIMENSIONALI RESTITUITI

## STRALCI DEL CODICE REALBASIC

A testimonianza del lavoro di scrittura del codice *RealBasic* per lo sviluppo del *software Brunelleschi*, ne vengono di seguito riportati alcuni stralci (il 5% del codice complessivo). Si ricorda che, come da licenza d'uso, il codice sorgente è di proprietà esclusiva dello scrivente.

```
App.OpenDocument:  
Sub OpenDocument(item As FolderItem)
```

```
    aprirestituzionework(item)
```

```
End Sub
```

```
...
```

```
App.Close:  
Sub Close()
```

```
    Dim f As FolderItem  
    Dim outstream As TextOutputStream  
    dim a,b as integer
```

```
    f=getfolderItem("prefs.txt")
```

```
    if f.exists then  
        f.delete  
    end if
```

```
    outstream=f.CreateTextFile
```

```
    for a=0 to 3  
        for b=0 to 3  
            outstream.WriteLine str(attributilinee(a,b))  
        next  
    next  
    outstream.WriteLine str(immaginedepth)
```

```
    outstream.Close
```

```
End Sub
```

```
App.Open:  
Sub Open()
```

```
    Dim f as FolderItem  
    Dim instream as TextInputStream  
    dim s as string  
    dim a,b as integer
```

```
    s=""  
    f=getfolderItem("prefs.txt")
```

```
    if f.Exists then  
        instream = f.OpenAsTextFile  
        for a=0 to 3  
            for b=0 to 3  
                s=instream.readline  
                attributilinee(a,b)=val(s)  
            next  
        next  
        s=instream.readline  
        immaginedepth=val(s)  
        instream.Close  
    else  
        ripristinapreferenze  
    end if
```

```
    iniziodazero
```

```
    scrx=screen(0).width  
    scry=screen(0).height
```

```
End Sub
```

```
fin_ori.aggiornabottoniorientamento:  
Sub aggiornabottoniorientamento()
```

```
    dim a as integer
```

```
    for a=0 to 8  
        Bev_ori(a).enabled=false  
    next  
    EditFi(0).enabled=true  
    EditFi(1).enabled=true
```

```

select case momento
case 100
  EditFi(0).enabled=false
  EditFi(1).enabled=false
  if zoomsiono then
    Bev_ori(6).enabled=true
  else
    if zoom_work<>1 then
      Bev_ori(7).enabled=true
    end if
    Bev_ori(8).enabled=true
  end if
  rect_scelto.left=-100
  if puntodivista(0,0)=0 then
    Bev_ori(0).enabled=true
  else
    PushButton1.enabled=true
  end if
  if orizzonte(0)>0 then
    for a=1 to 5
      Bev_ori(a).enabled=true
    next
  end if
else
  eliminapuntielinee
end select

attivoannulla=false
enableMenuItems

End Sub

fin_ori.sceltacomandoori:
Sub sceltacomandoori(index as integer)

  rect_scelto.width=bev_ori(index).width+6
  rect_scelto.height=bev_ori(index).height+6
  rect_scelto.left=bev_ori(index).left-3
  rect_scelto.top=bev_ori(index).top-3
  rect_puoisel.left=-200

  comando=110+(index*10)
  momento=comando

  aggiornabottoniorientamento
  aggiornainfo
  testoclip=fin_aiuto.aiuto_testo.text
  attivoannulla=true
  if fin_work.immacerchio<>nil then
    fin_work.backdrop=fin_work.immacerchio
    fin_work.immacerchio=nil
  end if
  'fin_work.refresh

select case comando

```

```

case 120,130,140,150,160
  fin_work.posiziona_orizzonte
  fin_work.posiziona_fondamentale
case 170
  fittowindowvista
  momento=100
  comando=0
  aggiornabottoniorientamento
  aggiornainfo
case 180
  visdimensionereale
end select

End Sub

fin_ori.puoiscoglierecomando:
Sub puoiscoglierecomando(index as integer)
  rect_puoisel.width=bev_ori(index).width+2
  rect_puoisel.height=bev_ori(index).height+2
  rect_puoisel.left=bev_ori(index).left-1
  rect_puoisel.top=bev_ori(index).top-1
End Sub

...

fin_ori.Bev_ori.MouseExit:
Sub MouseExit(Index As Integer)
  me.mouseCursor=arrowCursor
  rect_puoisel.left=-200
  fin_aiuto.aiuto_testo.text=testoclip
  fin_aiuto.link_testo.visible=true
End Sub

fin_ori.Bev_ori.MouseEnter:
Sub MouseEnter(Index As Integer)
  testoclip=fin_aiuto.aiuto_testo.text
  if me.enabled and comando=0 then
    me.mouseCursor=mano
    puoiscoglierecomando(index)
    fin_aiuto.aiuto_testo.text=me.helpTag
    fin_aiuto.link_testo.visible=false
  end if
End Sub

fin_ori.Bev_ori.Action:
Sub Action(Index As Integer)
  sceltacomandoori(index)
End Sub

...

fin_ori.EditFi.MouseExit:
Sub MouseExit(Index As Integer)
  fin_aiuto.aiuto_testo.text=testoclip
  fin_aiuto.link_testo.visible=true

```

End Sub

```
fin_ori.EditFi.MouseEnter:
Sub MouseEnter(Index As Integer)
  testoclip=fin_aiuto.aiuto_testo.text
  if me.enabled then
    fin_aiuto.aiuto_testo.text=me.helpTag
    fin_aiuto.link_testo.visible=false
  end if
End Sub
```

```
fin_ori.EditFi.KeyDown:
Function KeyDown(Index As Integer, Key As
String) As Boolean
  dim x,y as integer
  select case asc(key)
  case 13
    x=val(editFi(0).text)
    y=val(editFi(1).text)
    editFi(0).text=str(x)
    editFi(1).text=str(y)
    fin_work.puntoscelto(x,y)
    'fin_work.refresh
  case 27
    fin_work.annullacomando
  end select
End Function
```

...

```
fin_ori.PushButton1.Action:
Sub Action()
  if fin_work.immacerchio<>nil then
    fin_work.backdrop=fin_work.immacerchio
    fin_work.immacerchio=nil
  end if
  eliminapuntielinee
  self.close
  fin_tools.show
End Sub
```

...

```
fin_aiuto.assegnavalori:
Private Sub assegnavalori()
  dim f as integer
  if radioBu(1).value then
    f=1000
  else
    f=1
  end if
  editFi(0).setFocus
  editFi(1).setFocus
  x0=val(editFi(0).text)
  y0=val(editFi(1).text)
```

```
if x0*y0=0 then
  beep
else
  mostra_fields
  m0=y0/x0
  scalareale(0)=scalareale(0)+1
  redim scalareale(scalareale(0))
  scalareale(scalareale(0))=x0/f
  select case comando
  case 120
    fin_work.dis_pv120
  case 130
    fin_work.dis_pv130
  case 160
    if lineamq(3,4)=0 and lineamq(3,6)=0 then
      'ortogonale al quadro
      fin_work.dis_pv160a
    else
      'non ortogonale al quadro
      fin_work.dis_pv160b
    end if
  end select
end if
End Sub
```

```
fin_aiuto.assegnapoligono:
Private Sub assegnapoligono()
```

End Sub

```
fin_aiuto.assegnaduplicati:
Private Sub assegnaduplicati()
```

```
  dim f as integer
  editFi(3).setFocus
  f=val(editFi(3).text)
  editFi(3).text=str(f)
  if f<1 or f>64 then
    f=max(f,1)
    f=min(f,64)
    editFi(3).text=str(f)
    beep
  else
    mostra_fields
    rfp(1)=rfp(0)
    rfl(1)=rfl(0)
    rfl3d(1)=rfl3d(0)
    rfogg(1)=rfogg(0)
    fin_work.puntocomuneduplicati
    momento=momento+1
  end if
```

End Sub

```
fin_aiuto.assegnagriglia:  
Private Sub assegnagriglia()
```

```
    dim g1,g2 as integer  
    dim b as boolean
```

```
    editFi(0).setFocus  
    editFi(1).setFocus
```

```
    g1=val(editFi(0).text)  
    g2=val(editFi(1).text)  
    editFi(0).text=str(g1)  
    editFi(1).text=str(g2)  
    b=true
```

```
    if g1<3 or g2>64 then  
        g1=max(g1,2)  
        g1=min(g1,64)  
        editFi(0).text=str(g1)  
        b=false  
    end if
```

```
    if g2<3 or g2>64 then  
        g2=max(g2,2)  
        g2=min(g2,64)  
        editFi(1).text=str(g2)  
        b=false  
    end if
```

```
    if b then  
        mostra_fields  
        rfp(1)=rfp(0)  
        rfl(1)=rfl(0)  
        rfl3d(1)=rfl3d(0)  
        rfogg(1)=rfogg(0)  
        fin_work.disegnagriglia  
    else  
        beep  
    end if
```

```
End Sub
```

```
...
```

```
fin_aiuto.EditFi.KeyDown:  
Function KeyDown(Index As Integer, Key As  
String) As Boolean
```

```
    dim x,y as integer
```

```
    if me.top=22 then  
        select case asc(key)  
        case 13  
            select case comando  
            case 120  
                assegnavalori  
            case 1900
```

```
                assegnapoligono  
            case 5100  
                assegnaduplicati  
            case 5200  
                assegnagriglia  
            end select  
        case 27  
            mostra_fields  
            fin_work annullacomando  
        end select  
    end if
```

```
End Function
```

```
fin_aiuto.PushBu.Action:  
Sub Action(Index As Integer)
```

```
    select case index  
    case 0  
        assegnavalori  
    case 1  
        select case comando  
        case 1900  
            assegnapoligono  
        case 5200  
            assegnagriglia  
        end select  
    case 2  
        assegnaduplicati  
    end select  
End Sub
```

```
...
```

```
globali_app.importaimmagine:  
Sub importaimmagine()
```

```
    Dim f as FolderItem
```

```
    f=GetOpenFolderItem("special/any")  
    If f <> nil then  
        lineamq(0,0)=0  
        puntoxy(0,0)=0  
        fin_work.close  
        fin_ori.close  
        'fin_particolare.close  
        azzerorientamento  
        immagineimporta(f)  
        fin_ori.show  
        'fin_particolare.show  
        momento=100  
        aggiornainfo  
    end if
```

```
End Sub
```

...

globali\_app.azzerorientamento:

Sub azzerorientamento()

lineamq(0,1)=0

puntoxy(0,1)=0

orizzonte(0)=0

puntodivista(0,0)=0

scalareale(0)=0

screale=1

oo=0

pvx=0

pvy=0

fin\_ori.StaticT(3).text="?"

fin\_ori.StaticT(4).text="?"

fin\_ori.StaticT(5).text="?"

fin\_ori.StaticT(6).text="1"

fin\_ori.StaticT(8).visible=false

fin\_ori.StaticT(9).visible=false

fin\_ori.StaticT(10).visible=false

fin\_ori.PushButton1.enabled=false

rfl(0)=0

rfp(0)=0

rfl3d(0)=0

rfogg(0)=0

linea3Dxyz(0,0)=0

oggetto3d(0,0)=0

attivoannulla=false

End Sub

globali\_app.ingrandiscivista:

Sub ingrandiscivista()

dim x1,y1,x2,y2 as integer

dim maxWidth, maxHeight, xx as integer

dim dx,dy as integer

dim imma as picture

fin\_work.visible=false

zoomsiono=true

x1=min(fin\_work.punto(1).left+3,fin\_work.punto(2).left+3)

x2=max(fin\_work.punto(1).left+3,fin\_work.punto(2).left+3)

y1=min(fin\_work.punto(1).top+3,fin\_work.punto(2).top+3)

y2=max(fin\_work.punto(1).top+3,fin\_work.punto(2).top+3)

dx=(x2-x1)/zoom\_work

dy=(y2-y1)/zoom\_work

dx\_work=x1/zoom\_work

dy\_work=y1/zoom\_work

maxWidth = scrx-148

maxHeight = scry-112

zoom\_work = Min( maxWidth / dx,

maxHeight / dy )

imma = NewPicture(dx,dy,imaginedepth)

imma.graphics.drawPicture imma,

0,0,imma.width,imma.height,dx\_work,

dy\_work,imma.width,imma.height

visualizzaimma(imma)

mostranascondi(viselementi,false)

End Sub

globali\_app.visualizzaimma:

Sub visualizzaimma(imma as picture)

Dim p as picture

Dim zoomnometemp As Integer

dim xx as integer

dim xxx as double

p = NewPicture( imma.Width \* zoom\_work,

imma.Height \* zoom\_work, imaginedepth )

p.graphics.DrawPicture imma,

0,0,p.width,p.height,

0,0,imma.width,imma.height

fin\_work.width=p.width

fin\_work.height=p.height

fin\_work.backdrop=p

zoomnometemp=zoom\_work\*100

fin\_work.title=nomedelfileimma+" -

("+str(zoomnometemp)+"%)"

fin\_work.visible=true

#If TargetMacOS

fin\_work.top=110

#Elseif TargetWin32

fin\_work.top=107

#EndIf

fin\_work.left=146

if ff=0 then

ff=(fin\_work.height+0)/zoom\_work

xx=ff\*1000000

xxx=xx/1000000

fin\_ori.staticT(2).text=str(xxx)

end if

End Sub

globali\_app.fittowindowvista:

Sub fittowindowvista()

dim maxWidth, maxHeight, xx as integer

```

fin_work.visible=false
zoomsiono=false
maxWidth = scrx-148
maxHeight = scry-112
zoom_work = Min( maxWidth /
immagine.Width, maxHeight /
immagine.Height )
dx_work=0
dy_work=0
visualizzaimmagine(immagine)
mostranascondi(viselementi,false)

```

End Sub

globali\_app.ripristinapreferenze:

Sub ripristinapreferenze()

```

dim a as integer
attributilinee(0,0)=255
attributilinee(0,1)=0
attributilinee(0,2)=0
attributilinee(0,3)=1
attributilinee(1,0)=0
attributilinee(1,1)=255
attributilinee(1,2)=0
attributilinee(1,3)=1
attributilinee(2,0)=0
attributilinee(2,1)=255
attributilinee(2,2)=255
attributilinee(2,3)=1
attributilinee(3,0)=0
attributilinee(3,1)=0
attributilinee(3,2)=255
attributilinee(3,3)=1
for a=0 to 3

```

```

colore(a)=rgb(attributilinee(a,0),attributilinee(a,
1),attributilinee(a,2))

```

```

next

```

```

immaginedepth=24

```

End Sub

globali\_app.dimensionereale:

Sub dimensionereale()

```

dx_work=0
dy_work=0
fin_work.visible=false
zoomsiono=true
eliminapuntielinee
zoom_work = 1
visualizzaimmagine(immagine)
mostranascondi(viselementi,false)

```

End Sub

globali\_app.eliminapuntielinee:

Sub eliminapuntielinee()

```

dim a as integer

```

```

if fin_work.testorect.left>-100 then

```

```

    fin_work.testorect.left=-90

```

```

end if

```

```

fin_work.xyz3dtesto.left=-100

```

```

'fin_work.title=str(puntoxy(0,0))

```

```

'if fin_work.punto(1)<>nil then 'and

```

```

fin_work.linea(1)<>nil then

```

```

    for a=(rfp(0)+1) to puntoxy(0,0)

```

```

        fin_work.punto(a).visible=false

```

```

        'fin_work.punto(a).refresh

```

```

    next

```

```

for a=(rfl(0)+1) to lineamq(0,0)

```

```

    fin_work.linea(a).visible=false

```

```

    'fin_work.linea(a).refresh

```

```

next

```

```

'end if

```

```

puntoxy(0,1)=rfp(0)

```

```

lineamq(0,1)=rfl(0)

```

End Sub

globali\_app.registrarestituzionework:

Sub registrarestituzionework(f as folderItem)

```

Dim outstream As BinaryStream

```

```

dim a,b as integer

```

```

if f.exists then

```

```

    f.delete

```

```

end if

```

```

'registra impostazioni

```

```

outstream=f.createBinaryFile("brunelleschi/wor
kfile")

```

```

    outstream.Write

```

```

fileimmagine.absolutePath+chr(13)

```

```

    outstream.Write protezione(0)+chr(13)

```

```

    outstream.Write protezione(1)+chr(13)

```

```

for a=0 to 3

```

```

    for b=0 to 3

```

```

        outstream.Write

```

```

str(attributilinee(a,b))+chr(13)

```

```

    next

```

```

next

```

```

outstream.Write str(immaginedepth)+chr(13)

outstream.Write str(ff)+chr(13)

outstream.Write str(orizzonte(0))+chr(13)
for a=1 to orizzonte(0)
  outstream.Write str(orizzonte(a))+chr(13)
next
outstream.Write str(oo)+chr(13)

outstream.Write str(scalareale(0))+chr(13)
for a=1 to scalareale(0)
  outstream.Write str(scalareale(a))+chr(13)
next
outstream.Write str(screale)+chr(13)

outstream.Write str(puntodivista(0,0))+chr(13)
for a=1 to puntodivista(0,0)
  outstream.Write
str(puntodivista(a,0))+chr(13)
  outstream.Write
str(puntodivista(a,1))+chr(13)
next
outstream.Write str(pvx)+chr(13)
outstream.Write str(pvy)+chr(13)

outstream.Write str(momento)+chr(13)

outstream.Write str(lineamq(0,0))+chr(13)
for a=1 to lineamq(0,0)
  for b=0 to 8
    outstream.Write str(lineamq(a,b))+chr(13)
  next
next

outstream.Write str(linea3Dxyz(0,0))+chr(13)
for a=1 to linea3Dxyz(0,0)
  for b=0 to 7
    outstream.Write str(linea3Dxyz(a,b))+chr(13)
  next
next

outstream.Write str(oggetto3d(0,0))+chr(13)
for a=1 to oggetto3d(0,0)
  for b=0 to 4
    outstream.Write str(oggetto3d(a,b))+chr(13)
  next
next

'chiude
outstream.Close

End Sub

globali_app.aprirestituizionework:

```

```

Sub aprirestituizionework(f as folderItem)

  Dim g as folderItem
  Dim instream as TextInputStream
  dim s as string
  dim a,b as integer

  if f<>nil then

    instream = f.OpenAsTextFile

    ...

    If fileimmagine<>Nil Then
      if fileimmagine.exists then
        'nulla
      else
        msgBox "Immagine raster non
trovata,"+chr(13)+"procedere con la ricerca
manuale.."
        g=GetOpenFolderItem("special/any")
        If g <> nil and g.name=fileimmagine.name
then
          fileimmagine=g
        else
          fileimmagine=nil
        end if
      end if
    else
      msgBox "Immagine raster non
trovata,"+chr(13)+"procedere con la ricerca
manuale.."
      g=GetOpenFolderItem("special/any")
      If g <> nil then
        fileimmagine=g
      end if
    end if

    instream.Close

    if fileimmagine<>nil then

      instream = f.OpenAsTextFile

      s=instream.readline
      s=instream.readline
      s=instream.readline

      fin_work.close

      for a=0 to 3
        for b=0 to 3
          s=instream.readline
          attributilinee(a,b)=val(s)
        next
      next
    end if
  end if
End Sub

```

```

next
s=instream.readline
immaginedepth=val(s)

s=instream.readline
ff=val(s)

s=instream.readline
orizzonte(0)=val(s)
redim orizzonte(orizzonte(0))
for a=1 to orizzonte(0)
  s=instream.readline
  orizzonte(a)=val(s)
next
s=instream.readline
oo=val(s)

s=instream.readline
scalareale(0)=val(s)
redim scalareale(scalareale(0))
for a=1 to scalareale(0)
  s=instream.readline
  scalareale(a)=val(s)
next
s=instream.readline
screale=val(s)

s=instream.readline
puntodivista(0,0)=val(s)
redim puntodivista(puntodivista(0,0),1)
for a=1 to puntodivista(0,0)
  s=instream.readline
  puntodivista(a,0)=val(s)
  s=instream.readline
  puntodivista(a,1)=val(s)
next
s=instream.readline
pvx=val(s)
s=instream.readline
pvy=val(s)

s=instream.readline
momento=val(s)

s=instream.readline
lineamq(0,0)=val(s)
redim lineamq(lineamq(0,0),8)
for a=1 to lineamq(0,0)
  for b=0 to 8
    s=instream.readline
    lineamq(a,b)=val(s)
  next
next

s=instream.readline

```

```

linea3Dxyz(0,0)=val(s)
redim linea3Dxyz(linea3Dxyz(0,0),7)
for a=1 to linea3Dxyz(0,0)
  for b=0 to 7
    s=instream.readline
    linea3Dxyz(a,b)=val(s)
  next
next

rf13d(0)=linea3Dxyz(0,0)
rf13d(1)=linea3Dxyz(0,0)

s=instream.readline
oggetto3d(0,0)=val(s)
redim oggetto3d(oggetto3d(0,0),4)
for a=1 to oggetto3d(0,0)
  for b=0 to 4
    s=instream.readline
    oggetto3d(a,b)=val(s)
  next
next

rfogg(0)=oggetto3d(0,0)
rfogg(1)=oggetto3d(0,0)

'apparecchia
for a=0 to 3

colore(a)=rgb(attributilinee(a,0),attributilinee(a,
1),attributilinee(a,2))
next
immagineimporta(fileimmagine)
if momento=100 then
  fin_tools.close
  fin_ori.show
else
  fin_ori.close
  fin_tools.show
  disegnalineedafile
end if
fileapertoosalvato=f
aggiornainfo

instream.Close
else
  MsgBox "Immagine raster non trovata o non
utilizzabile."
end if

end if

End Sub

globali_app.immagineimporta:

```

```

Sub immagineimporta(f as folderItem)

    dim p as picture
    dim maxWidth, maxHeight as integer
    dim a,l,b,c as integer

    nomedelfileimmagine=f.displayName
    p = f.OpenAsPicture
    if p<>nil then
        fileimmagine=f

        b=fileimmagine.Length/100

        l=len(fileimmagine.name)
        for a=1 to l
            b=b+asc(mid(fileimmagine.name,a,1))
        next

        c=((b+111)/2)+sqrt(b/2)+251

        immagine = NewPicture( p.Width, p.Height,
        immaginedepth )
        immagine.graphics.DrawPicture p, 0,0
    
```

...

End Sub

```

globali_app.visdimensionereale:
Sub visdimensionereale()

```

```

    dim maxWidth,maxHeight as integer
    dim f as double
    dim m as integer

    maxWidth=scrx-148
    maxHeight=scry-112

    eliminapuntielinee
    if immagine.width<maxWidth and
    immagine.height<maxHeight then
        dimensionereale
        aggiornainfo
        momento=100
        if momento<999 then
            fin_ori.aggiornabottoniorientamento
        else
            fin_tools.aggiornabottonitools
        end if
        comando=0
    else
        if momento<999 then
            fin_ori.aggiornabottoniorientamento
            momento=180
            comando=180
        
```

```

    else
        fin_tools.aggiornabottonitools
        momento=9000
        comando=9000
    end if
    aggiornainfo

    x0=min(maxwidth,immagine.width)*zoom_wor
    k

    y0=min(maxheight,immagine.height)*zoom_wor
    k
        fin_work.dis_punto(-10,-10,0)

    fin_work.dis_linea(1,1,attributilinee(0,3),colore(
    0))

    fin_work.dis_linea(1,1,attributilinee(0,3),colore(
    0))
        fin_work.dis_punto(-10,-10,0)

    fin_work.dis_linea(2,2,attributilinee(0,3),colore(
    0))

    fin_work.dis_linea(2,2,attributilinee(0,3),colore(
    0))
    end if

```

End Sub

```

globali_app.mostranascondi:
Sub mostranascondi(tipo as integer, metti as
boolean)

```

```

    dim a as integer
    dim b(2,1) as boolean

    b(0,0)=true
    b(0,1)=true
    b(1,0)=true
    b(1,1)=false
    b(2,0)=false
    b(2,1)=false

    if fin_work.testorect.left>-100 then
        fin_work.testorect.left=-90
    end if
    fin_work.xyz3dtesto.left=-100

    for a=1 to rfl(0)
        fin_work.linea(a).x1=(lineamq(a,0)-
        dx_work)*zoom_work
        fin_work.linea(a).y1=(lineamq(a,1)-
        dy_work)*zoom_work
    
```

```

    fin_work.linea(a).x2=(lineamq(a,2)-
dx_work)*zoom_work
    fin_work.linea(a).y2=(lineamq(a,3)-
dy_work)*zoom_work
    select case lineamq(a,8)
    case 1
        fin_work.linea(a).visible=false
    case 2
        fin_work.linea(a).visible=b(tipo,1)
    case 3
        fin_work.linea(a).visible=b(tipo,0)
    end select
next
End Sub

globali_app.coloradi:
Sub coloradi()

    dim a as integer

    for a=rfl(1) to rfl(0)

fin_work.linea(a).lineColor=colore(lineamq(a,8))
    if lineamq(a,8)=1 then
        fin_work.linea(a).visible=false
    end if
next
'fin_work.refresh

End Sub

globali_app.disegnalineedafile:
Sub disegnalineedafile()
    dim a,b,i,o,ini,fin as integer
    dim x1,y1,x2,y2 as double

    b=lineamq(0,0)
    lineamq(0,0)=0
    lineamq(0,1)=0
    puntoxy(0,0)=0
    puntoxy(0,1)=0

    o=0
    ini=0
    fin=0
    for a=1 to b
    if lineamq(a,8)>1 then
        puntoxy(0,1)=0
        x1=(lineamq(a,0)-dx_work)*zoom_work
        y1=(lineamq(a,1)-dy_work)*zoom_work
        x2=(lineamq(a,2)-dx_work)*zoom_work
        y2=(lineamq(a,3)-dy_work)*zoom_work
        fin_work.dis_punto(x1,y1,-5)
        fin_work.dis_punto(x2,y2,-5)

```

```

    fin_work.dis_linea2(1,2,attributilinee(lineamq(a,
8),3),colore(lineamq(a,8)))

    calcolamq(1,2,lineamq(0,0),lineamq(a,7),lineamq
(a,8))
        i=lineamq(a,7)
        if linea3Dxyz(i,7)<>o and ini<>0 then
            fin=lineamq(0,0)-1
            oggetto3d(o,4)=fin
            o=linea3Dxyz(i,7)
            ini=lineamq(0,0)
            oggetto3d(o,3)=ini
        end if
        if linea3Dxyz(i,7)<>o and ini=0 then
            o=linea3Dxyz(i,7)
            ini=lineamq(0,0)
            oggetto3d(o,3)=ini
        end if
    next
    if o>0 then
        fin=lineamq(0,0)
        oggetto3d(o,4)=fin
    end if

    rfp(0)=0
    rfp(1)=0
    rfl(0)=lineamq(0,0)
    rfl(1)=rfl(0)
    eliminapuntielinee

End Sub

globali_app.eliminaelemento:
Sub eliminaelemento(visibile as boolean)

    dim a as integer

        oggetto3d(oggettoselezionato,0)=-
oggetto3d(oggettoselezionato,0)
        for a=oggetto3d(oggettoselezionato,1) to
oggetto3d(oggettoselezionato,2)
            linea3Dxyz(a,6)=-linea3Dxyz(a,6)
        next
        for a=oggetto3d(oggettoselezionato,3) to
oggetto3d(oggettoselezionato,4)
            lineamq(a,8)=-lineamq(a,8)
        next
        fin_work.linea(a).lineColor=colore(abs(lineamq(
a,8)))
        fin_work.linea(a).visible=visibile
    next
    momento=1000
    comando=0

```

```

agganciosnap=0
fin_tools.aggiornabottonitools
fin_tools.aggiornasnap
ultimo_comando=1600

End Sub

globali_app.visualizzaparticolare:
Sub visualizzaparticolare()

    dim p as picture
    dim x,y as integer

    if immagine<>nil then
        x=dx_work+(xyzpro(0)/zoom_work)-69
        y=dy_work+(xyzpro(2)/zoom_work)-46
        p = NewPicture(138,92,immaginedepth)
        p.graphics.DrawPicture
        immagine,0,0,immagine.width,immagine.height
        ,x,y,immagine.width,immagine.height
        fin_particolare.backdrop=p
    end if

End Sub

finestrainfo.PushButton1.Action:
Sub Action()
    finestrainfo.Close
End Sub

fin_work.dis_punto:
Sub dis_punto(xx as double, yy as double, col as integer)

    Dim pun As punto

    puntoxy(0,1)=puntoxy(0,1)+1

    if puntoxy(0,1)>puntoxy(0,0) then
        pun=New punto
        puntoxy(0,0)=puntoxy(0,1)
        redim puntoxy(puntoxy(0,0),1)
    else
        if col<>-5 then
            punto(puntoxy(0,1)).visible=true
        end if
    end if

    if col<>-5 then
        punto(puntoxy(0,1)).fillcolor=colore(col)
        punto(puntoxy(0,1)).left=xx-3
        punto(puntoxy(0,1)).top=yy-3
    end if

```

```

puntoxy(puntoxy(0,1),0)=dx_work+(xx/zoom_work)

puntoxy(puntoxy(0,1),1)=dy_work+(yy/zoom_work)

End Sub

fin_work.dis_linea:
Sub dis_linea(index1 as integer, index2 as integer, spe as integer, col as color)

    Dim lin As linea

    lineamq(0,1)=lineamq(0,1)+1

    if lineamq(0,1)>lineamq(0,0) then
        lin=New linea
        lineamq(0,0)=lineamq(0,1)
        redim lineamq(lineamq(0,0),8)
    else
        linea(lineamq(0,1)).visible=true
    end if

    linea(lineamq(0,1)).borderWidth=spe
    linea(lineamq(0,1)).linecolor=col
    linea(lineamq(0,1)).x1=(puntoxy(index1,0)-dx_work)*zoom_work+punto(index1).left+3
    linea(lineamq(0,1)).y1=(puntoxy(index1,1)-dy_work)*zoom_work+punto(index1).top+3
    linea(lineamq(0,1)).x2=(puntoxy(index2,0)-dx_work)*zoom_work+punto(index2).left+3
    linea(lineamq(0,1)).y2=(puntoxy(index2,1)-dy_work)*zoom_work+punto(index2).top+3

End Sub

fin_work.dis_orizzonte:
Sub dis_orizzonte()

    dim testoricevuto as string
    dim a as integer
    dim xx as integer
    dim xxx as double

    testoricevuto=calcolaintersezione(1,2)

    if testoricevuto<>"incidenti" then
        msgBox "Impossibile determinare l'intersezione!"+chr(13)+"Le due rette individuate sono "+testoricevuto+"."
    else
        orizzonte(0)=orizzonte(0)+1
        redim orizzonte(orizzonte(0))
    end if

```

```

orizzonte(orizzonte(0))=y0
oo=0
for a=1 to orizzonte(0)
  oo=oo+orizzonte(a)
next
oo=oo/orizzonte(0)
xx=oo*1000000
xxx=xx/1000000
fin_ori.StaticT(3).text=str(xxx)
fin_ori.StaticT(8).visible=true
for a=1 to 2
  if pow(lineamq(rfl(0)+a,1)-
orizzonte(orizzonte(0)),2)>pow(lineamq(rfl(0)+a
,3)-orizzonte(orizzonte(0)),2) then
    linea(rfl(0)+a).x2=(x0-
dx_work)*zoom_work
    linea(rfl(0)+a).y2=(y0-
dy_work)*zoom_work
  else
    linea(rfl(0)+a).x1=(x0-
dx_work)*zoom_work
    linea(rfl(0)+a).y1=(y0-
dy_work)*zoom_work
  end if
end if
next
dis_punto(-10,(orizzonte(orizzonte(0))-
dy_work)*zoom_work,1)

dis_punto(fin_work.width+10,(orizzonte(orizzo
nte(0))-dy_work)*zoom_work,1)
dis_linea(5,6,attributilinee(0,3),colore(0))
dis_punto(-10,(oo-dy_work)*zoom_work,1)
dis_punto(fin_work.width+10,(oo-
dy_work)*zoom_work,1)
dis_linea(7,8,attributilinee(1,3),colore(1))
end if

momento=100
comando=0

fin_ori.aggiornabottoniorientamento

```

End Sub

```

fin_work.puntoscelto110:
Private Sub puntoscelto110(x as integer, y as
integer)
  dim a as integer
  select case momento
  case 110
    dis_punto(x,y,0)
    dis_linea(1,1,attributilinee(0,3),colore(0))
    momento=momento+1
  case 111

```

```

    if x=punto(1).left+3 and y=punto(1).top+3
  then
    beep
  else
    dis_punto(x,y,0)
    calcolamq(1,2,rfl(0)+1,0,0)
    momento=momento+1
  end if
  case 112
    dis_punto(x,y,0)
    dis_linea(3,3,attributilinee(0,3),colore(0))
    momento=momento+1
  case 113
    if x=punto(3).left+3 and y=punto(3).top+3
  then
    beep
  else
    dis_punto(x,y,0)
    momento=momento+1
    calcolamq(3,4,rfl(0)+2,0,0)
    dis_orizzonte
  end if
  end select
aggiornainfo
End Sub

```

```

fin_work.puntoscelto:
Sub puntoscelto(x as double, y as double)

```

```

  dim a as boolean

```

```

  select case comando

```

```

  case 110
    puntoscelto110(x,y)
  case 120
    puntoscelto120(x,y)
  case 130
    puntoscelto130(x,y)
  case 140
    puntoscelto140(x,y)
  case 150
    puntoscelto150(x,y)
  case 160
    puntoscelto160(x,y)
  case 180

```

```

  puntoscelto180(linea(rfl(0)+1).x1/zoom_work,li
nea(rfl(0)+1).y1/zoom_work)
  case 190
    puntoscelto190(x,y)
  case 1300
    puntoscelto1300(x,y)
  case 1400
    resti_o_costru=0
    puntoscelto1400(x,y)

```

```

case 1500
  resti_o_costru=1
  puntosculto1400(x,y)
case 1700
  resti_o_costru=1
  puntosculto1700(x,y)
case 1800
  resti_o_costru=1
  puntosculto1800(x,y)
case 2600
  resti_o_costru=1
  puntosculto1400(x,y)
case 5000
  resti_o_costru=0
  puntosculto5000(x,y)
case 5100
  resti_o_costru=0
  puntosculto5100(x,y)
case 9000

puntosculto9000(linea(rfl(0)+1).x1 / zoom_work,
linea(rfl(0)+1).y1 / zoom_work)
end select
aggiornainfo

```

End Sub

```

fin_work.puntosculto190:
Private Sub puntosculto190(x as integer, y as
integer)
  dim a as integer
  select case momento
  case 190
    dis_punto(x,y,0)
    dis_linea(1,1,attributilinee(0,3),colore(0))
    dis_linea(1,1,attributilinee(0,3),colore(0))
    dis_linea(1,1,attributilinee(0,3),colore(0))
    dis_linea(1,1,attributilinee(0,3),colore(0))
    momento=momento+1
  case 191
    if x=punto(1).left+3 or y=punto(1).top+3 then
      beep
    else
      dis_punto(x,y,0)
      momento=momento+1
      ingrandiscivista
      eliminapuntielinee
      momento=100
      comando=0
      fin_ori.aggiornabottoniorientamento
    end if
  end select
  aggiornainfo
End Sub

```

```

fin_work.puntosculto180:
Private Sub puntosculto180(x as integer, y as
integer)

```

```

  dim dx,dy as integer
  dim imma as picture

```

```

  maxWidth = scrx-148
  maxHeight = scry-112
  dx_work=x
  dy_work=y
  dx=min(maxwidth,immagine.width)
  dy=min(maxheight,immagine.height)
  dx=min(immagine.width-x,dx)
  dy=min(immagine.height-y,dy)
  imma = NewPicture(dx,dy,immaginedepth)
  imma.graphics.drawPicture immagine,
0,0,immagine.width,immagine.height,x,y,imma
  gine.width,immagine.height
  fin_work.visible=false
  zoomsiono=true
  eliminapuntielinee
  zoom_work = 1
  visualizzaimmagine(imma)
  eliminapuntielinee
  momento=100
  fin_ori.aggiornabottoniorientamento
  comando=0
  aggiornainfo

```

End Sub

```

fin_work annullacomando:
Sub annullacomando()

```

```

  dim a as integer

```

```

  if attivoannulla then
    attivoannulla=false
    if ultimo_comando=5000 then
      attivoannulla=true
    end if
    select case momento
    case is<1000
      momento=100
      comando=0
      fin_ori.aggiornabottoniorientamento
      eliminapuntielinee
    case 1601
      for a=oggetto3d(oggettoselezionato,3) to
oggetto3d(oggettoselezionato,4)
        linea(a).lineColor=colore(lineamq(a,8))
      next
    else
      puntoxy(0,1)=0

```

```

lineamq(0,1)=rfl(1)
linea3Dxyz(0,0)=rfl3d(1)
oggetto3d(0,0)=rfogg(1)
rfl(0)=0
rfl(0)=rfl(1)
rfl3d(0)=rfl3d(1)
rfogg(0)=rfogg(1)
end select
if momento=1000 then
select case ultimo_comando
case 5000
muovielementoannulla
attivoannulla=false
case 1600
eliminaelemento(true)
end select
end if
if momento>999 then
momento=1000
comando=0
eliminapuntielinee
fin_tools.aggiornabottonitools
fin_tools.aggiornasnap
end if
aggiornainfo
end if

End Sub

fin_work.puntoscelto120:
Private Sub puntoscelto120(x as integer, y as
integer)
dim a,b as integer
select case momento
case 120
dis_punto(x,y,0)
dis_linea(5,5,attributilinee(0,3),colore(0))
momento=momento+1
case 121
if linea(rfl(0)+3).x2=punto(5).left+3 then
beep
else
dis_punto(linea(rfl(0)+3).x2,linea(rfl(0)+3).y2,0)
calcolamq(5,6,rfl(0)+3,0,0)
dis_linea(6,6,attributilinee(0,3),colore(0))
momento=momento+1
end if
case 122
b=0
if x=punto(5).left+3 and y=punto(5).top+3
then
b=1
end if
if y=punto(6).top+3 then

```

```

b=1
end if
if b=1 then
beep
else
dis_punto(x,y,0)
momento=momento+1
calcolamq(6,7,rfl(0)+4,0,0)
fin_aiuto.mostra_fields
end if
end select
aggiornainfo
End Sub

fin_work.dis_pv120:
Sub dis_pv120()

dim testoricevuto as string
dim a,b as integer
dim xx as integer
dim x1,x2,xxx as double

testoricevuto=calcolaintersezione(1,4)
x1=x0

dis_linea(5,7,attributilinee(0,3),colore(0))
calcolamq(5,7,rfl(0)+5,0,0)
testoricevuto=calcolaintersezione(1,5)
x2=x0

dis_punto((x1-dx_work)*zoom_work,(oo-
dy_work)*zoom_work,0)
dis_punto((x1-dx_work)*zoom_work,(oo-
abs(m0*(x1-x2))-dy_work)*zoom_work,0)
dis_punto((x2-dx_work)*zoom_work,(oo-
dy_work)*zoom_work,0)
dis_linea(8,9,attributilinee(0,3),colore(0))
dis_linea(9,10,attributilinee(0,3),colore(0))
dis_linea(5,8,attributilinee(0,3),colore(0))
linea(rfl(0)+5).x2=(x2-dx_work)*zoom_work
linea(rfl(0)+5).y2=(oo-dy_work)*zoom_work

testoricevuto=calcolaintersezione(2,4)
dis_punto((x0-dx_work)*zoom_work,(y0-
dy_work)*zoom_work,0)
linea(rfl(0)+4).x1=(x1-dx_work)*zoom_work
linea(rfl(0)+4).y1=(oo-dy_work)*zoom_work
linea(rfl(0)+4).x2=(x0-dx_work)*zoom_work
linea(rfl(0)+4).y2=(y0-dy_work)*zoom_work
q0=x0
calcolamq(5,8,rfl(0)+8,0,0)
testoricevuto=calcolaintersezione(2,8)
dis_punto((x0-dx_work)*zoom_work,(y0-
dy_work)*zoom_work,0)
linea(rfl(0)+8).x1=(x1-dx_work)*zoom_work

```

```

linea(rfl(0)+8).y1=(oo-dy_work)*zoom_work
linea(rfl(0)+8).x2=(x0-dx_work)*zoom_work
linea(rfl(0)+8).y2=(y0-dy_work)*zoom_work

failamediadeipuntidivista(x1,abs(m0*(x1-
x2)),13)

scalareale(scalareale(0))=scalareale(scalareale(0)
)/abs(q0-x0)
failamediadellascala(scalareale(scalareale(0)))

momento=100
comando=0

fin_ori.aggiornabottoniorientamento
aggiornainfo

End Sub

fin_work.posiziona_orizzonte:
Sub posiziona_orizzonte()
dis_punto(-10,(oo-dy_work)*zoom_work,-5)
dis_punto(fin_work.width+10,(oo-
dy_work)*zoom_work,-5)

dis_linea(rfp(0)+1,rfp(0)+2,attributilinee(1,3),col
ore(1))
calcolamq(rfp(0)+1,rfp(0)+2,rfl(0)+1,0,0)
End Sub

fin_work.muovilinea:
Private Sub muovilinea(x1 as integer, y1 as
integer, x2 as integer, y2 as integer, index as
integer)
linea(index).x1=x1
linea(index).y1=y1
linea(index).x2=x2
linea(index).y2=y2
End Sub

fin_work.posiziona_fondamentale:
Sub posiziona_fondamentale()

dis_punto(-10,(ff-dy_work)*zoom_work,-5)
dis_punto(fin_work.width+10,(ff-
dy_work)*zoom_work,-5)

dis_linea(rfp(0)+3,rfp(0)+4,attributilinee(1,3),col
ore(1))
calcolamq(rfp(0)+3,rfp(0)+4,rfl(0)+2,0,0)

End Sub

fin_work.puntoscelto1300:

```

```

Private Sub puntoscelto1300(x as integer, y as
integer)
dim a as integer
select case momento
case 1300
dis_punto(x,y,0)
dis_linea(1,1,attributilinee(0,3),colore(0))
dis_linea(1,1,attributilinee(0,3),colore(0))
dis_linea(1,1,attributilinee(0,3),colore(0))
dis_linea(1,1,attributilinee(0,3),colore(0))
momento=momento+1
case 1301
if x=punto(1).left+3 or y=punto(1).top+3 then
beep
else
dis_punto(x,y,0)
momento=momento+1
ingrandiscivista
eliminapuntielinee
momento=1000
comando=0
fin_tools.aggiornabottonitools
ultimo_comando=1300
end if
end select
aggiornainfo
End Sub

fin_work.posiziona_pppd:
Sub posiziona_pppd()
dis_punto((pvx-dx_work)*zoom_work,(oo-
dy_work)*zoom_work,1)
dis_punto((pvx+pvy-
dx_work)*zoom_work,(oo-
dy_work)*zoom_work,1)
End Sub

fin_work.puntoscelto9000:
Private Sub puntoscelto9000(x as integer, y as
integer)

dim dx,dy as integer
dim imma as picture

maxWidth = scrx-148
maxHeight = scry-112
dx_work=x
dy_work=y
dx=min(maxwidth,immagine.width)
dy=min(maxheight,immagine.height)
dx=min(immagine.width-x,dx)
dy=min(immagine.height-y,dy)
imma = NewPicture(dx,dy,immaginedepth)

```

```

imma.graphics.drawPicture immagine,
0,0,immagine.width,immagine.height,x,y,imma
gine.width,immagine.height
fin_work.visible=false
zoomsiono=true
eliminapuntielinee
zoom_work = 1
visualizzaimmagine(imma)
eliminapuntielinee
momento=1000
comando=0
agganciosnap=0
fin_tools.aggiornabottonitools
fin_tools.aggiornasnap
aggiornainfo
mostranascondi(viselementi,false)

```

End Sub

```

fin_work.muoviassixyz:
Sub muoviassixyz(x as double, y as double)

```

```

dim xx,yy,h1,h2,xf1,yf1,yff as integer
dim xxx,yyy as double
dim testoricevuto as string
dim b,c as boolean

b=visibili_assi
if keyboard.asyncCommandKey and
attivocommand then
h1=xyzpro(1)
h2=y
c=true
else
h1=-10
h2=self.height+10
xyzpro(0)=x
if agganciosnap<>6 and agganciosnap<>4
then
xyzpro(1)=y
end if
c=false
end if
if agganciosnap=6 then
c=true
end if
controllasnaptasti(x,y,0,0,0)

```

```

xx=(dx_work+(xyzpro(0)/zoom_work))*100000
0

```

```

yy=(dy_work+(xyzpro(1)/zoom_work))*100000
0
xxx=xx/1000000

```

```

yyy=yy/1000000
fin_tools.EditFi(0).text=str(xyzpro(0))
fin_tools.EditFi(1).text=str(xyzpro(1))
fin_tools.staticT(0).text=str(xxx)
fin_tools.staticT(1).text=str(yyy)

```

```

eliminapuntielinee
posiziona_orizzonte
posiziona_fondamentale
posiziona_pppd

```

```

if visibili_assi or c then

```

```

if y>linea(rfl(0)+1).top or c then

```

```

dis_punto(xyzpro(0),y,-5)
dis_punto(-10,y,-5)
dis_punto(self.width+10,y,-5)

```

```

dis_linea(rfp(0)+8,rfp(0)+9,attributilinee(1,3),col
ore(1))

```

```

dis_punto(xyzpro(0),h1,-5)
dis_punto(xyzpro(0),h2,-5)

```

```

dis_linea(rfp(0)+10,rfp(0)+11,attributilinee(1,3),c
olore(1))

```

```

if xyzreale(0,0)=xyzreale(0,1) and
xyzreale(1,0)=xyzreale(1,1) then
b=false
end if

```

```

dis_linea(rfp(0)+5,rfp(0)+7,attributilinee(1,3),col
ore(1))

```

```

if y>linea(rfl(0)+1).top then
yff=fin_work.height-punto(rfp(0)+5).top-3
else
yff=-punto(rfp(0)+5).top-3
end if

```

```

xf1=punto(rfp(0)+5).left+3-xyzpro(0)
yf1=y-punto(rfp(0)+5).top-3
xf1=yff*xf1/yf1
yf1=yff+punto(rfp(0)+5).top+3
linea(rfl(0)+5).x2=punto(rfp(0)+5).left+3-xf1
linea(rfl(0)+5).y2=yf1

```

```

if not visibili_assi and c then

```

```

b=true
end if
linea(rfl(0)+3).visible=visibili_assi
linea(rfl(0)+4).visible=b
linea(rfl(0)+5).visible=visibili_assi
testorect.visible=visibili_assi
xyz3dtesto.visible=visibili_assi

```

```

if y>linea(rfl(0)+1).top then

```

```

    xyz3dtesto.top=y+7
else
    xyz3dtesto.top=y-50
end if

if xyzpro(0)>punto(rfp(0)+5).left+3 then
    xyz3dtesto.textAlign=2
    xyz3dtesto.left=xyzpro(0)-80
else
    xyz3dtesto.textAlign=0
    xyz3dtesto.left=xyzpro(0)+10
end if
if testorect.left>-100 then
    testorect.left=xyz3dtesto.left-5
    testorect.top=xyz3dtesto.top
end if

else
    if testorect.left>-100 then
        testorect.left=-90
    end if
    xyz3dtesto.left=-100
end if

end if

End Sub

fin_work.puntoscelto1400:
Private Sub puntoscelto1400(x as double, y as double)

    dim a as integer

    select case momento
    case 1400,1500,2600
        rfp(1)=rfp(0)
        rfl(1)=rfl(0)
        rfl3d(1)=rfl3d(0)
        rfogg(1)=rfogg(0)
        puntocomune1(x,y,true)
        attivocommand=true
        attivoshift=true
        forzaquota=false
        forzadirezione=false
    case 1401,1501
        a=momento-1
        oggetto3d(0,1)=momento
        puntocomune2(x,y,true)
        inseriscioggetto3d
        ultimo_comando=a
    case 2601
        a=momento-1
        oggetto3d(0,1)=momento
        puntocomune2(x,y,true)

```

```

        dis_completapiano
        inseriscioggetto3d
        ultimo_comando=a
    end select
    aggiornainfo

End Sub

fin_work.controllasnaptasti:
Sub controllasnaptasti(x as double, y as double,
i as integer, k as integer, f as integer)

    dim b,v as boolean
    dim xx,yy,zz,ll as integer
    dim xxx,yyy,zzz,lll as double

daprospeffivaareale(xyzpro(0),xyzpro(1),xyzpro(2))

    xyzreale(0,2)=0
    xyzreale(1,2)=0
    xyzreale(2,2)=0

    select case agganciosnap
    case 4

        darealeaprospeffiva(xyzreale(0,0),xyzreale(1,1),xyzreale(2,0))
        xyzpro(0)=x
        xyzreale(1,2)=1
    case 5
        if keyboard.asyncCommandKey and attivocommand then
            '
        else
            darealeaprospeffiva(xyzreale(0,1),xyzreale(1,0),xyzreale(2,0))
            xyzpro(1)=y
            xyzreale(0,2)=1
        end if
    case 6

        darealeaprospeffiva(xyzreale(0,1),xyzreale(1,1),xyzreale(2,0))
        xyzreale(0,2)=1
        xyzreale(1,2)=1
    end select

    b=false
    if keyboard.asyncShiftKey and attivoshift then
        b=true
    end if
    if forzaquota then

```

```

b=true
end if

if b then

darealeaprospettiva(xyzreale(0,0),xyzreale(1,0),x
yzreale(2,1))
  if xyzreale(0,2)=0 then
    xyzpro(0)=x
  end if
  if xyzreale(1,2)=0 then
    xyzpro(1)=y
  end if
  xyzreale(2,2)=1
else
  if xyzreale(2,2)=0 then
    xyzpro(2)=y
  end if
end if

if forzadirezione then

daprospettivaareale(xyzpro(0),xyzpro(1),xyzpro
(2))
  if not verticalexz then
    if not verticalexy then
      xyzreale(1,0)=(m0*xyzreale(0,0))+q0
    if not attivocommand then
      xyzreale(2,0)=(m1*xyzreale(0,0))+q1
    end if
    xx=sign(x-punto(2).left)*sign(xq0)
  else
    xyzreale(0,0)=xyzreale(0,1)
  if not attivocommand then
    xyzreale(2,0)=(m1*xyzreale(1,0))+q1
  end if
  xx=sign(y-punto(2).top)*sign(yq0)
end if
else
  xyzreale(0,0)=m1
  xyzreale(1,0)=q1
  xyzpro(2)=y
end if
if b then
  xyzreale(2,0)=xyzreale(2,1)
end if

darealeaprospettiva(xyzreale(0,0),xyzreale(1,0),x
yzreale(2,0))
end if

if forzaquadrato and right(str(momento),1)="3"
then
  xyzreale(0,0)=xyzreale(0,1)+(xx*xq0)
  xyzreale(1,0)=xyzreale(1,1)+(xx*yq0)

```

```

darealeaprospettiva(xyzreale(0,0),xyzreale(1,0),x
yzreale(2,1))
end if

if forzaaltezza and right(str(momento),1)="5"
then
  ll=lunghezza(1)*sign(punto(2).top-y)

darealeaprospettiva(xyzreale(0,0),xyzreale(1,0),x
yzreale(2,1)+ll)
end if

if xyzreale(0,2)=1 then
  xyzreale(0,0)=xyzreale(0,1)
end if
if xyzreale(1,2)=1 then
  xyzreale(1,0)=xyzreale(1,1)
end if
if xyzreale(2,2)=1 then
  xyzreale(2,0)=xyzreale(2,1)
end if

if not b and not keyboard.asyncCommandKey
and agganciosnap<>6 and agganciosnap<>4
then
  xyzreale(2,0)=0
end if

lunghezza(0)=sqrt(pow(xyzreale(0,0)-
xyzreale(0,1),2)+pow(xyzreale(1,0)-
xyzreale(1,1),2)+pow(xyzreale(2,0)-
xyzreale(2,1),2))

lunghezza(0)=lunghezza(0)*sign(val(right(str(m
omento),1)))
xx=xyzreale(0,0)*1000000
yy=xyzreale(1,0)*1000000
zz=xyzreale(2,0)*1000000
ll=lunghezza(0)*1000000
xxx=xx/1000000
yyy=-yy/1000000
zzz=zz/1000000
lll=ll/1000000
xyz3dtesto.text="X = "+str(xxx)+chr(13)+"Y =
"+str(yyy)+chr(13)+"Z = "+str(zzz)+chr(13)+"L =
"+str(lll)

End Sub

fin_work.puntocomune1:
Private Sub puntocomune1(x as double, y as
double, check as boolean)

```

```

dim b as boolean

```

```

b=true

if linea(rfl(0)+1)<>nil and check then
  if xyzpro(1)<linea(rfl(0)+1).top then
    b=false
  end if
end if

if b then
  eliminapuntielinee
  dis_punto(xyzpro(0),xyzpro(1),1)
  dis_punto(xyzpro(0),xyzpro(2),0)
  dis_punto(xyzpro(0),xyzpro(1),-5)
  dis_punto(xyzpro(0),xyzpro(2),-5)

daprospectivaareale(xyzpro(0),xyzpro(1),xyzpro(2))
  xyzreale(0,1)=xyzreale(0,0)
  xyzreale(1,1)=xyzreale(1,0)
  xyzreale(2,1)=xyzreale(2,0)

dis_linea(rfp(0)+1,rfp(0)+3,attributilinee(2,3),colore(1))

dis_linea(rfp(0)+2,rfp(0)+4,attributilinee(0,3),colore(0))
  momento=momento+1
  rfp(0)=rfp(0)+4
  rfl(0)=rfl(0)+2
  agganciosnap=0
  fin_tools.aggiornasnap
else
  beep
end if

```

End Sub

```

fin_work.puntocomune2:
Private Sub puntocomune2(x as double, y as double, ultimo as boolean)

```

```

  dim col(1) as integer

  if xyzpro(1)>linea(rfl(0)+1).top then

    if not forzasnappunto then
      controllasnaptasti(x,y,0,0,0)
    else
      forzasnappunto=false
    end if

    muoviquattrolinee
    momento=momento+1

```

```

  puntoxy(rfp(0)-
1,0)=dx_work+(xyzpro(0)/zoom_work)
  puntoxy(rfp(0)-
1,1)=dy_work+(xyzpro(1)/zoom_work)

puntoxy(rfp(0),0)=dx_work+(xyzpro(0)/zoom_work)

puntoxy(rfp(0),1)=dy_work+(xyzpro(2)/zoom_work)
  inseriscilinea3d(1,resti_o_costru,rfogg(0)+1)
  col(1)=2+resti_o_costru
  rfl3d(0)=rfl3d(0)+1
  col(0)=2
  calcolamq(rfp(0)-3,rfp(0)-1,rfl(0)-
1,linea3Dxyz(0,0),col(0))
  calcolamq(rfp(0)-
2,rfp(0),rfl(0),linea3Dxyz(0,0),col(1))
  rfp(0)=0
  coloradi
  eliminapuntielinee
  lunghezza(1)=lunghezza(0)
  if ultimo then
    momento=1000
    comando=0
    agganciosnap=0
    fin_tools.aggiornabottonitools
    fin_tools.aggiornasnap
  end if

else
  beep
end if

```

End Sub

```

fin_work.puntoscelto1800:
Private Sub puntoscelto1800(x as double, y as double)

```

```

  select case momento
  case 1800
    rfp(1)=rfp(0)
    rfl(1)=rfl(0)
    rfl3d(1)=rfl3d(0)
    rfogg(1)=rfogg(0)
    puntocomune1(x,y,true)
    attivocommand=false
    attivoshift=false
    forzaquota=true
    forzadirezione=false
  case 1801
    puntocomune2(x,y,false)
    lunghezza(1)=lunghezza(0)
    calcolamqrealiperpendicolari(rfl3d(0))

```

```

puntocomune1(x,y,true)
fin_tools.disabilitasnap
attivocommand=false
attivoshift=false
forzaquota=true
forzadirezione=true
case 1803
puntocomune2(x,y,false)
dis_completaquadrato
puntocomune1(x,y,false)
fin_tools.disabilitasnap
attivocommand=false
attivoshift=false
forzaquota=false
forzadirezione=false
agganciosnap=6
case 1805
oggetto3d(0,1)=momento
puntocomune2(x,y,true)
dis_completaparallelepipedo
inseriscioggetto3d
ultimo_comando=1800
end select
aggiornainfo

End Sub

fin_work.dis_linea2:
Sub dis_linea2(index1 as integer, index2 as
integer, spe as integer, col as color)

Dim lin As linea

lineamq(0,1)=lineamq(0,1)+1

lin=New linea
lineamq(0,0)=lineamq(0,1)

linea(lineamq(0,1)).borderWidth=spe
linea(lineamq(0,1)).linecolor=col
linea(lineamq(0,1)).x1=(puntoxy(index1,0)-
dx_work)*zoom_work'punto(index1).left+3
linea(lineamq(0,1)).y1=(puntoxy(index1,1)-
dx_work)*zoom_work'punto(index1).top+3
linea(lineamq(0,1)).x2=(puntoxy(index2,0)-
dx_work)*zoom_work'punto(index2).left+3
linea(lineamq(0,1)).y2=(puntoxy(index2,1)-
dx_work)*zoom_work'punto(index2).top+3

End Sub

fin_work.dalinea3d_lineapro:
Sub dalinea3d_lineapro(index as integer, p1 as
integer)

```

```

dim a,b as integer
dim x1,y1,z1,x2,y2,z2 as double

darealeaprospettiva(linea3Dxyz(index,0),linea3
Dxyz(index,1),linea3Dxyz(index,2))
x1=xyzpro(0)
y1=xyzpro(1)
z1=xyzpro(2)

darealeaprospettiva(linea3Dxyz(index,3),linea3
Dxyz(index,4),linea3Dxyz(index,5))
x2=xyzpro(0)
y2=xyzpro(1)
z2=xyzpro(2)
if p1>0 then
puntoxy(0,1)=0
dis_punto(x1,y1,-5)
dis_punto(x2,y2,-5)
dis_linea(1,2,attributilinee(2,3),colore(2))
rfl(0)=rfl(0)+1
calcolamq(1,2,rfl(0),index,2)
end if
puntoxy(0,1)=0
dis_punto(x1,z1,-5)
dis_punto(x2,z2,-5)
dis_linea(1,2,attributilinee(3,3),colore(3))
rfl(0)=rfl(0)+1
calcolamq(1,2,rfl(0),index,3)
rfl(0)=0

End Sub

fin_work.dis_completaquadrato:
Private Sub dis_completaquadrato()

dim a as integer

'if linea3Dxyz(linea3Dxyz(0,0)-1,2)=0 then
'a=0
'else
a=1
'end if
linea3Dxyz(0,0)=linea3Dxyz(0,0)+1
redim linea3Dxyz(linea3Dxyz(0,0),7)

linea3Dxyz(linea3Dxyz(0,0),0)=linea3Dxyz(linea
3Dxyz(0,0)-1,3)

linea3Dxyz(linea3Dxyz(0,0),1)=linea3Dxyz(linea
3Dxyz(0,0)-1,4)

linea3Dxyz(linea3Dxyz(0,0),2)=linea3Dxyz(linea
3Dxyz(0,0)-1,5)

```

```
linea3Dxyz(linea3Dxyz(0,0),3)=linea3Dxyz(linea
3Dxyz(0,0),0)-linea3Dxyz(linea3Dxyz(0,0)-
2,3)+linea3Dxyz(linea3Dxyz(0,0)-2,0)
```

```
linea3Dxyz(linea3Dxyz(0,0),4)=linea3Dxyz(linea
3Dxyz(0,0),1)-linea3Dxyz(linea3Dxyz(0,0)-
2,4)+linea3Dxyz(linea3Dxyz(0,0)-2,1)
```

```
linea3Dxyz(linea3Dxyz(0,0),5)=linea3Dxyz(linea
3Dxyz(0,0),2)-linea3Dxyz(linea3Dxyz(0,0)-
2,5)+linea3Dxyz(linea3Dxyz(0,0)-2,2)
linea3Dxyz(linea3Dxyz(0,0),6)=1
linea3Dxyz(linea3Dxyz(0,0),7)=rfogg(0)+1
delinea3d_lineapro(linea3Dxyz(0,0),a)
linea3Dxyz(0,0)=linea3Dxyz(0,0)+1
redim linea3Dxyz(linea3Dxyz(0,0),7)
```

```
linea3Dxyz(linea3Dxyz(0,0),0)=linea3Dxyz(linea
3Dxyz(0,0)-1,3)
```

```
linea3Dxyz(linea3Dxyz(0,0),1)=linea3Dxyz(linea
3Dxyz(0,0)-1,4)
```

```
linea3Dxyz(linea3Dxyz(0,0),2)=linea3Dxyz(linea
3Dxyz(0,0)-1,5)
```

```
linea3Dxyz(linea3Dxyz(0,0),3)=linea3Dxyz(linea
3Dxyz(0,0)-3,0)
```

```
linea3Dxyz(linea3Dxyz(0,0),4)=linea3Dxyz(linea
3Dxyz(0,0)-3,1)
```

```
linea3Dxyz(linea3Dxyz(0,0),5)=linea3Dxyz(linea
3Dxyz(0,0)-3,2)
linea3Dxyz(linea3Dxyz(0,0),6)=1
linea3Dxyz(linea3Dxyz(0,0),7)=rfogg(0)+1
delinea3d_lineapro(linea3Dxyz(0,0),a)
rfl3d(0)=rfl3d(0)+2
eliminapuntielinee
```

End Sub

```
fin_work.dis_completaparallelepipedo:
Private Sub dis_completaparallelepipedo()
```

```
dim a,b1,b2 as integer
dim z as double
```

```
b1=linea3Dxyz(0,0)-4
b2=linea3Dxyz(0,0)-1
z=linea3Dxyz(linea3Dxyz(0,0),5)-
linea3Dxyz(linea3Dxyz(0,0),2)
```

```
for a=b1+1 to b2
```

```
linea3Dxyz(0,0)=linea3Dxyz(0,0)+1
redim linea3Dxyz(linea3Dxyz(0,0),7)
```

```
linea3Dxyz(linea3Dxyz(0,0),0)=linea3Dxyz(a,0)
```

```
linea3Dxyz(linea3Dxyz(0,0),1)=linea3Dxyz(a,1)
```

```
linea3Dxyz(linea3Dxyz(0,0),2)=linea3Dxyz(a,2)
```

```
linea3Dxyz(linea3Dxyz(0,0),3)=linea3Dxyz(a,0)
```

```
linea3Dxyz(linea3Dxyz(0,0),4)=linea3Dxyz(a,1)
```

```
linea3Dxyz(linea3Dxyz(0,0),5)=linea3Dxyz(a,2)+
z
```

```
linea3Dxyz(linea3Dxyz(0,0),6)=1
linea3Dxyz(linea3Dxyz(0,0),7)=rfogg(0)+1
delinea3d_lineapro(linea3Dxyz(0,0),0)
```

```
next
```

```
for a=b1 to b2
```

```
linea3Dxyz(0,0)=linea3Dxyz(0,0)+1
redim linea3Dxyz(linea3Dxyz(0,0),7)
```

```
linea3Dxyz(linea3Dxyz(0,0),0)=linea3Dxyz(a,0)
```

```
linea3Dxyz(linea3Dxyz(0,0),1)=linea3Dxyz(a,1)
```

```
linea3Dxyz(linea3Dxyz(0,0),2)=linea3Dxyz(a,2)+
z
```

```
linea3Dxyz(linea3Dxyz(0,0),3)=linea3Dxyz(a,3)
```

```
linea3Dxyz(linea3Dxyz(0,0),4)=linea3Dxyz(a,4)
```

```
linea3Dxyz(linea3Dxyz(0,0),5)=linea3Dxyz(a,5)+
z
```

```
linea3Dxyz(linea3Dxyz(0,0),6)=1
linea3Dxyz(linea3Dxyz(0,0),7)=rfogg(0)+1
delinea3d_lineapro(linea3Dxyz(0,0),0)
```

```
next
```

```
rfl3d(0)=rfl3d(0)+7
```

```
eliminapuntielinee
```

End Sub

```
fin_work.tempmessaggio:
Private Sub tempmessaggio(index as integer)
```

```
dim i,o as integer
dim l as double
dim s as string
```

```
fin_lunghezza.show
i=lineamq(index,7)
```

```

l=sqrt(pow(linea3Dxyz(i,3)-
linea3Dxyz(i,0),2)+pow(linea3Dxyz(i,4)-
linea3Dxyz(i,1),2)+pow(linea3Dxyz(i,5)-
linea3Dxyz(i,2),2))
s="Lunghezza segmento reale: "+str(l)
fin_lunghezza.staticTe1.text=s
o=linea3Dxyz(i,7)
s="Dati generali elemento:"+chr(13)+chr(13)
s=s+"Segmento reale n. "+str(i)+"; (costruzione
"+str(linea3Dxyz(i,6))+")"+chr(13)+chr(13)
s=s+"Segmento prospettico n. "+str(index)+";
(colore
"+str(lineamq(index,8))+")"+chr(13)+chr(13)
s=s+"Elemento restituito n. "+str(o)+chr(13)
s=s+"Tipo (id): "+str(oggetto3d(o,0))+chr(13)
s=s+"Seg.3d primo:
"+str(oggetto3d(o,1))+chr(13)
s=s+"Seg.3d ultimo:
"+str(oggetto3d(o,2))+chr(13)
s=s+"Seg.pr. primo:
"+str(oggetto3d(o,3))+chr(13)
s=s+"Seg.pr. ultimo: "+str(oggetto3d(o,4))
fin_lunghezza.staticTe2.text=s

```

End Sub

fin\_work.muoviquattrolinee:

Private Sub muoviquattrolinee()

```

if xyzpro(1)>linea(rfl(0)+1).top then
  muovilinea(punto(rfp(0)-
3).left+3,punto(rfp(0)-
3).top+3,xyzpro(0),xyzpro(1),rfl(0)-1)
  muovilinea(punto(rfp(0)-
2).left+3,punto(rfp(0)-
2).top+3,xyzpro(0),xyzpro(2),rfl(0))
end if

```

End Sub

fin\_work.muovielemento:

Sub muovielemento(sposta as boolean)

```

dim a,i as integer
dim l3d(8),x1,y1,z1,x2,y2,z2 as double

```

```

l3d(6)=xyzpro(0)
l3d(7)=xyzpro(1)
l3d(8)=xyzpro(2)
m_xyz(0,1)=xyzreale(0,0)
m_xyz(1,1)=xyzreale(1,0)
m_xyz(2,1)=xyzreale(2,0)

```

for a=oggetto3d(oggettoselezionato,3) to  
oggetto3d(oggettoselezionato,4)

```

i=lineamq(a,7)
l3d(0)=linea3Dxyz(i,0)+m_xyz(0,1)-
m_xyz(0,0)
l3d(1)=linea3Dxyz(i,1)+m_xyz(1,1)-
m_xyz(1,0)
l3d(2)=linea3Dxyz(i,2)+m_xyz(2,1)-
m_xyz(2,0)
l3d(3)=linea3Dxyz(i,3)+m_xyz(0,1)-
m_xyz(0,0)
l3d(4)=linea3Dxyz(i,4)+m_xyz(1,1)-
m_xyz(1,0)
l3d(5)=linea3Dxyz(i,5)+m_xyz(2,1)-
m_xyz(2,0)
darealeaprospettiva(l3d(0),l3d(1),l3d(2))
x1=xyzpro(0)
y1=xyzpro(1)
z1=xyzpro(2)
darealeaprospettiva(l3d(3),l3d(4),l3d(5))
x2=xyzpro(0)
y2=xyzpro(1)
z2=xyzpro(2)
linea(a).x1=x1
linea(a).x2=x2
if sposta then
  lineamq(a,0)=(x1 / zoom_work)+dx_work
  lineamq(a,2)=(x2 / zoom_work)+dx_work
end if
select case lineamq(a,8)
case 3
  linea(a).y1=z1
  linea(a).y2=z2
  if sposta then
    lineamq(a,1)=(z1 / zoom_work)+dy_work
    lineamq(a,3)=(z2 / zoom_work)+dy_work
  end if
case 2
  linea(a).y1=y1
  linea(a).y2=y2
  if sposta then
    lineamq(a,1)=(y1 / zoom_work)+dy_work
    lineamq(a,3)=(y2 / zoom_work)+dy_work
  end if
end select
if sposta then

```

...

## BIBLIOGRAFIA

Anna Sgrosso, *Note di Fotogrammetria Applicata all'Architettura*, Napoli, 1979

Mario Docci, Diego Maestri, *Scienza del Disegno*, Torino, 2000

Riccardo Migliari, *Fondamenti della Rappresentazione Geometrica e Informatica dell'Architettura*, Roma, 2000

Andrea Pozzo, *Prospettiva de' Pittori e Architetti*, Roma, 1693 (edizione a cura di Maria Walcher Casotti, Trieste, 2003)

Marco Canciani, *Il Disegno in Prospettiva*, Roma, 2005

Mario Docci, Diego Maestri, *Manuale di Rilevamento Architettonico e Urbano*, Roma-Bari, 2005

Jerry Lee Ford, *Beginning REALbasic*, Berkeley (California), 2006

Mark S. Choate, *REALbasic Cross-Platform Application Development*, Indianapolis (Indiana), 2006