

Edició i simulació de il·luminació en obertures d'edificis

Albert Mas, Gonzalo Besuievsky

Grup de Gràfics de Girona
Universitat de Girona

Febrer 2005

Resum

El disseny arquitectònic tendeix cada cop més a emprar eines de software diferents als productes CAD convencionals, buscant obtenir altres resultats que permetin donar un valor afegit al producte final. Una d'aquestes tendències és la generació de models arquitectònics tridimensionals a partir de models bidimensionals, on els poden observar les estructures dissenyades des d'un altre punt de vista, oferint més vistositat al projecte i permetent al dissenyador comprovar la validesa del seu disseny. Degut a la dificultat que implica aquest tipus de generació, una eina que pogués obtenir de forma ràpida i senzilla models tridimensionals a partir dels dissenys en planta permetria estalviar molts recursos en temps i costos.

La il·luminació en entorns virtuals té un rol fonamental en el disseny arquitectònic, tant per a la visualització del propi entorn com per a el disseny d'un sistema d'il·luminació adequat. El dimensionat eficient de les obertures del edifici permet aprofitar millor la il·luminació solar, fet que ofereix un estalvi energètic i de costos en sistemes de il·luminació. Una eina que ofereixi ajudes per aquestes tasques permetria establir de forma correcta la il·luminació de les estructures dissenyades, alhora que podria generar imatges de qualitat dels entorns virtuals.

Paraules clau: CAD arquitectònic, generació 3D, il·luminació solar, entorn virtual

1 Introducció

El disseny arquitectònic tendeix cada cop més a emprar eines de software diferents a les oferides per els productes CAD convencionals. Aquesta tendència té com objectiu obtenir un altre tipus de resultats en base a un projecte arquitectònic, els quals permetin donar un valor afegit al producte final. S'han analitzat dos necessitats del sector: la generació de models 3D i la simulació de il·luminació natural.

El procés de disseny arquitectònic es centra principalment en el modelat 2D de la planta i/o alçats. Un cop finalitzats pot sorgir la necessitat de generar el model 3D de tota l'edificació, i per tant cal dissenyar de nou tot el model. Una eina que obtingui de forma ràpida i senzilla models tridimensionals a partir dels dissenys en planta permetria estalviar molts recursos en temps i costos. Alhora oferiria la possibilitat de validació visual de distribució de volums, i atorgaria un valor afegit en la presentació final del projecte.

Un cop el model 3D està generat, la il·luminació aplicada a aquest adquireix un rol fonamental, tant per a una correcta visualització com per a el disseny d'un sistema d'il·luminació adequat. El correcte dimensionat de les obertures permet aprofitar millora la il·luminació solar, fet que permet estalviar energia i costos en el sistema d'il·luminació artificial. Una eina que ofereixi ajudes per aquestes tasques permetria establir de forma eficient la il·luminació de les estructures dissenyades, alhora que podria generar imatges vistoses dels entorns virtuals com a valor afegit al projecte del arquitecte.

1.1 Eines actuals

S'han analitzat diverses eines actuals en funció dels objectius plantejats. De totes elles s'han discretitzat dos categories: CADs arquitectònics i eines de modelat 3D, junt amb les seves respectives capacitats de renderitzat i simulació de il·luminació.

1.1.1 Eines de CAD arquitectònic

S'han escollit tres eines representatives del sector: AutoCAD, Architectural Desktop i AllPlan.

AutoCAD de Autodesk [Aut97], és potser el software més estès i emprat del mercat. El seu domini de treball és el modelat 2D, sense moltes facilitats per al modelat 3D. El procés per crear una planta en 3D és totalment manual i difícilment es poden aprofitar els plànols originals. La simulació de il·luminació és molt limitada i poc eficient.

Architectural Desktop de Autodesk [Arc01], és el següent pas evolutiu de AutoCAD. Conserva totes les seves funcionalitats, i afegeix la possibilitat de dissenyar objectes en 2D i 3D alhora, tal i que mentre s'emplacen elements vectorials en 2D en el plànol, també s'estan definint els respectius objectes 3D. Però la seva limitació es centra en la necessitat d'emprar l'eina des del primer pas de disseny del projecte, obligant a un procés merament manual convertir plànols 2D creats amb altres programes a models 3D. La simulació de il·luminació millora sensiblement, tot i que segueix estant limitada.

AllPlan de Nemetscheck, reuneix les funcionalitats dels productes abans comentats, i tot i que ofereix algunes eines per a la generació semi-automàtica de models 3D a partir de plànols 2D,

encara és una tasca lenta i manual. De la mateixa forma que els altres, les capacitats de simulació d'il·luminació són limitades.

1.1.2 Eines de modelat 3D

Aquest tipus d'eines és d'ús comú per els dissenyadors que busquen crear models 3D. En referència al modelat 3D no hi han moltes diferències entre els productes analitzats, basant-se aquestes en un procés manual de construcció d'objectes. Sí que es poden observar diferències en la simulació d'il·luminació. Productes com Maya, d'Alias—Wavefront, permeten simular la il·luminació solar o emprar nous i recents sistemes de rendering com el Photon Mapping. També el VIZ, de Autodesk, o el Accurender, de Robert McNeel, permeten calcular la il·luminació solar, i permeten el renderitzat amb mètodes de càlcul de la radiositat.

1.2 Objectius

En funció dels objectius inicials i les observacions realitzades sobre productes existents s'ha arribat a les següents conclusions:

- No existeixen eines que permetin el pas automàtic de models 2D arquitectònics a models 3D.
- Molts productes permeten el càlcul de la il·luminació natural, però es basen en processos costosos computacionalment, sense moltes possibilitats de interactivitat.

Així doncs, aquest treball es centra en la solució a aquests dos problemes. Per un costat s'aborda un mètode capaç d'oferir automàticament models 3D a partir de models 2D, o si més no amb una interactivitat mínima per part del usuari. Per altra banda s'estudia un sistema que simula la il·luminació natural sobre el model 3D generat, i que permet visualitzar i modificar la geometria de forma interactiva junt amb els canvis d'il·luminació respectius. Aquesta geometria a modificar es centra en l'edició de les mesures de les obertures, així com la simulació de il·luminació a diferents hores del dia.

2 Generació del model 3D

Un disseny arquitectònic 2D sol estar format per informació vectorial, la qual en el seu conjunt representa elements com parets, portes, finestres o símbols [AuxDXF]. El nostre sistema és capaç de classificar aquesta informació a partir d'una estructura inherent en el procés de disseny: l'assignació de capes de treball. Aquestes capes solen estructurar-se en funció del tipus d'element a representar, i cada element vectorial està associat a una capa. Un cop identificats els elements 2D i les seves característiques, la generació dels models corresponents 3D és automàtica.

Després d'analitzar les dades incloses en dissenys 2D, es poden definir tres línies bàsiques d'estudi per a la generació d'una estructura 3D bàsica d'una planta: parets, portes i finestres, i sostre i terra.

2.1 Parets

El mètode definit per a la generació de les parets es basa en un algorisme de tres passos. El primer pas identifica els elements vectorials que les representen, especificant una o mes capes de treball. Aquests elements estan formats per línies i/o polilínies. El segon pas consisteix en trobar els perfils de les parets a partir d'aquests elements vectorials, tal i que siguin polígons tancats i orientats positivament (veure Fig.1b). La característica de polígon tancat i orientat positivament ens permet especificar el interior i exterior de la paret [FDFHP], i per tant determinar els vectors normals de cada polígon 3D que seran creats en el darrer pas.

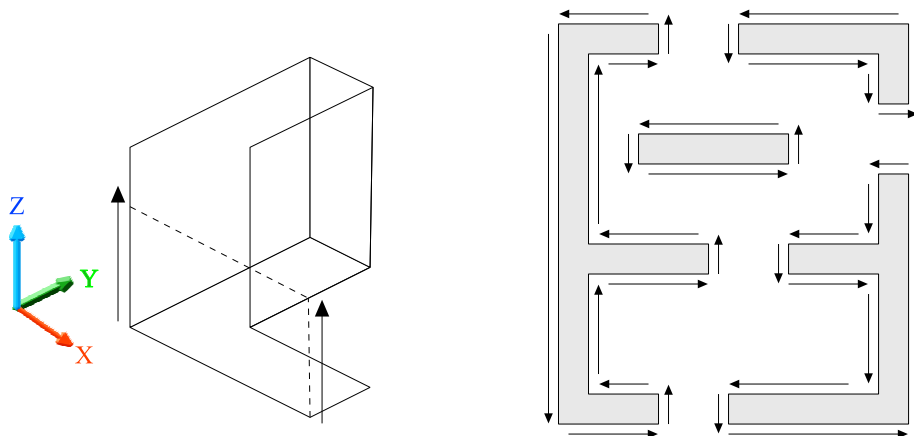


Figura 1: A l'esquerra (a) la generació de les cares que formaran les parets. A la dreta (b) es mostra les seqüències de línies trobades que formen polígons tancats orientats positivament.

S'han considerat dos mètodes de representació 2D de les parets per trobar aquests perfils. El primer assumeix que les línies i polilínies representen el perfil de les parets, amb el que dos línies paral·leles formen una paret. En aquest cas s'aplica un algorisme que busca seqüències tancades de línies i polilínies, i un cop trobades orienta positivament els seus vèrtexs. El segon assumeix que les parets són representades amb línies amb un cert gruix, i per tant una sola línia és suficient per representar una paret. En aquest cas s'aplica un algorisme que crea línies paral·leles a les originals i separades entre sí segons el gruix de cada línia original. Unint les noves línies i tancant els extrems s'aconsegueixen els polígons tancats buscats (veure Fig.2a). Finalment, tan sols queda orientar positivament els nous vèrtexs creats.

El tercer i darrer pas per a la generació de parets 3D es basa en la extrosió d'aquests polígons una longitud igual a l'alçada designada per a l'estructura, i en direcció perpendicular a la planta (veure Fig.1a i Fig.2b).

2.2 Portes i finestres

Les portes i finestres en un pla 2D es representen amb un símbol format per un conjunt d'elements vectorials. Però entre els dissenyadors no hi ha un consens en la utilització d'un grup de símbols estàndard aplicable al procés de disseny arquitectònic actual.

Amb aquesta limitació, s'ha dissenyat un algorisme de tres passos. El primer selecciona totes les portes i finestres a partir de la selecció dels identificadors que els representen, usualment basats en etiquetes [AuxDXF]. El segon pas determina les dos parets més properes al punt d'inserció del símbol (veure Fig.3). Utilitzant aquestes parets com a límits es genera un rectangle que ocupa l'espai entre les dos parets.

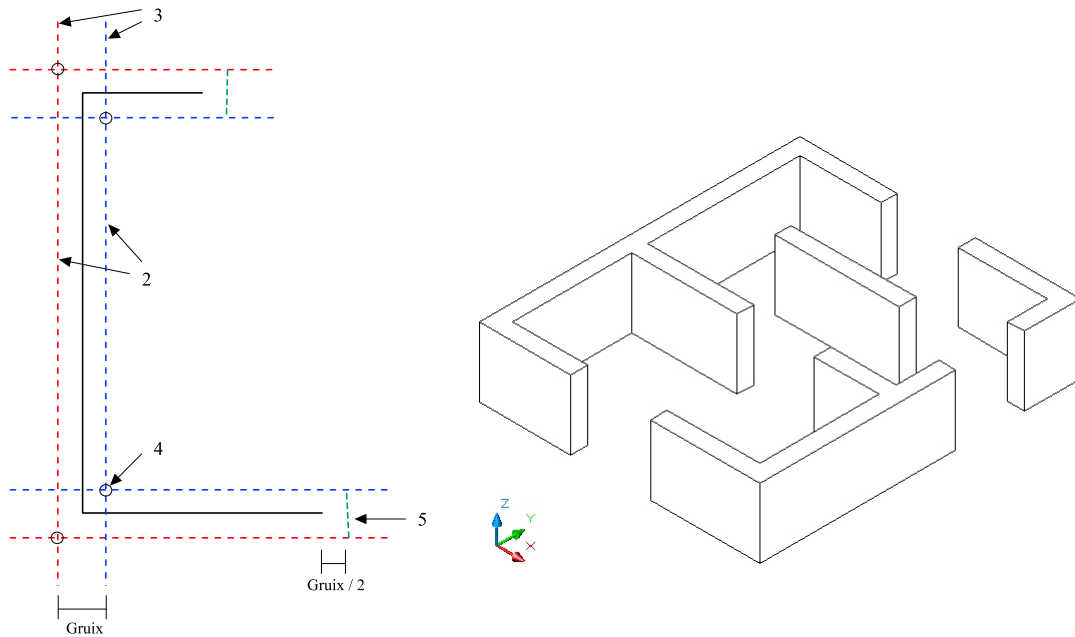


Figura 2: A l'esquerra (a) generació del perfil de la paret a partir d'una línia simple. A la dreta (b) el model 3D final de les parets.

El tercer pas genera l'objecte 3D a partir de la extrusió d'aquest rectangle en diferents etapes, segons si el tipus d'obertura és una finestra o una porta, i en la mateixa direcció que les parets (veure Fig.4).

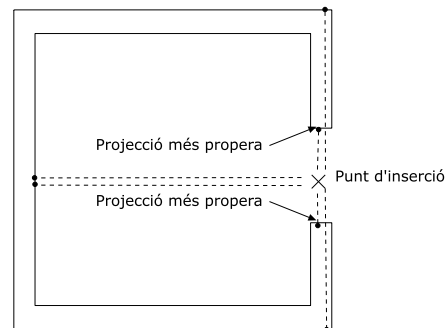


Figura 3: Emplaçament de les obertures.

2.3 Terra i sostre

L'objectiu de creació d'un terra i sostre sorgeix amb la necessitat d'acabar el model 3D, creant tapes en la seva part inferior i superior. El mètode més simple i ràpid és la generació de tapes a partir del envoltant rectangular dels perfils de les parets. Aquestes tapes es construeixen amb prismes de reduïda alçada, en lloc de simples plans (veure Fig.5), evitant així futurs problemes en diferents sistemes de visualització 3D [FDFHP].

L'emplaçament d'una tapa superior i inferior permet tancar els volums de les parets, portes i finestres. Però això pot produir un efecte indesitjat en determinats sistemes de il·luminació. El problema apareix en els marges inferiors i superiors de les parets, els quals incideixen sobre les tapes

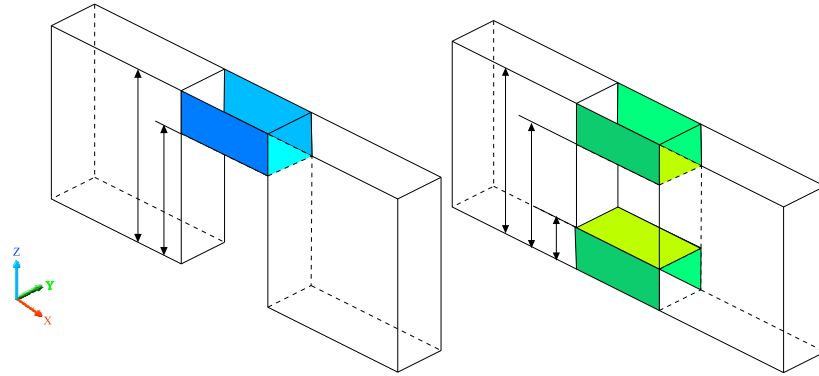


Figura 4: *Construcció de portes i finestres.*

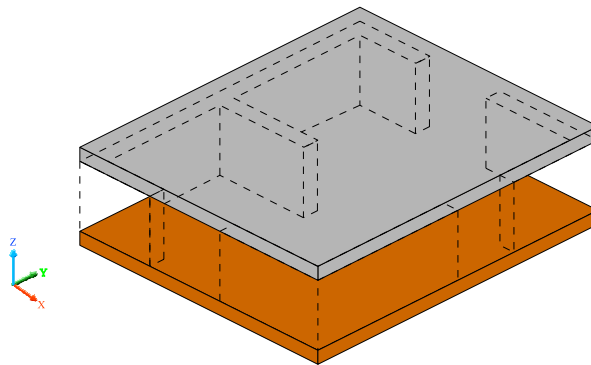


Figura 5: *Construcció del sostre i terra.*

creades. Es poden obtenir resultats incorrectes si el sistema de il·luminació emprat no considera aquestes unions, aplicant la il·luminació calculada sobre cada polígon segons la seva geometria. Fins i tot si s'utilitzen mètodes de discretització de superfícies, el problema persisteix (veure Fig.6). Aquest efecte es demostra més detalladament a [SP94].

La solució aportada, anomenada *Cerca d'habitacions*, permet identificar les zones interiors a les habitacions. L'algorisme parteix dels polígons 2D generats a partir del envolupant rectangular i dels perfils de les parets, portes i finestres. Realitzant una operació booleana de intersecció obtenim l'interior de les habitacions. En la Fig.7 s'observen els passos a seguir. Primer es cerca el envolupant rectangular de la planta. Després es calculen els polígons dels perfils de les parets més les zones cobertes per obertures. Finalment es realitza una operació booleana de diferència entre el envolupant rectangular i la seqüència de polígons. La configuració final de les tapes és la mateixa, però amb les bases modificades per representar tan sols les zones interiors a les habitacions (veure Fig.8).

2.4 Altres elements

S'ha afegit la possibilitat de crear i emplaçar altres elements, com mobiliari, fonts de llum artificials i replicació de plantes per formar un edifici. La inclusió de mobiliari és un procés més manual que implica escollir un moble d'una llibreria i inserir-lo en el disseny. El mateix succeeix amb les fonts de llum, amb la diferència que es poden escollir entre fonts de llum omnidireccionals, focus o d'àrea. En ambdós casos es pot emprar l'algorisme d'emplaçament d'obertures per emplaçar aquests items en funció dels símbols inclosos en el disseny. La replicació de plantes es basa en una repetició simple

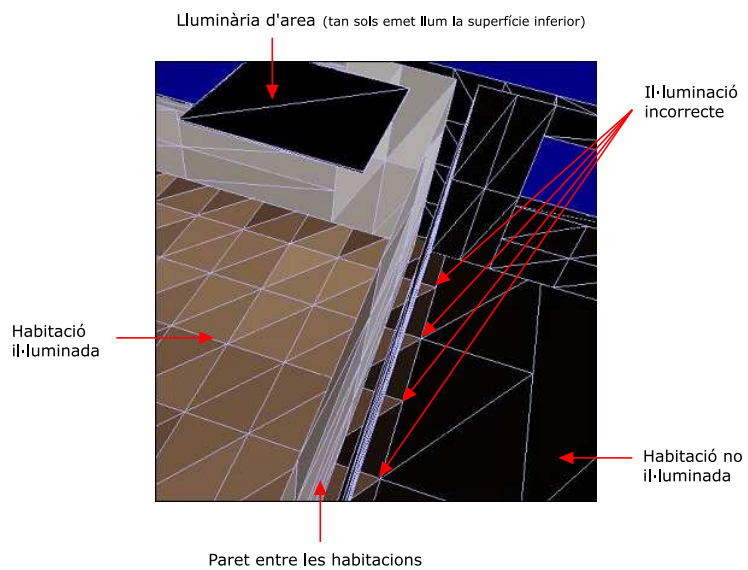


Figura 6: Problema en les unions de les parets amb el terra i sostre.

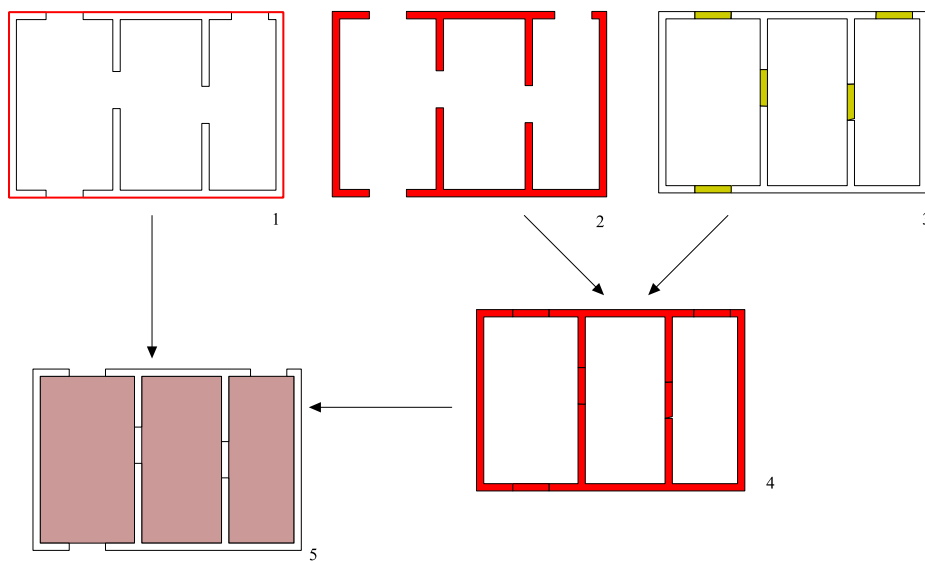


Figura 7: Algoritme Cerca d'habitacions.

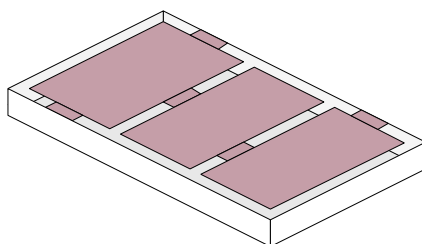


Figura 8: Construcció final del terra.

de la geometria amb tot el seu contingut, tal i que s'emplacen les plantes una sobre l'altre, formant un edifici.

3 Il·luminació solar

El càlcul de la il·luminació solar és complex. Intervenien molts factors, des de el dia i hora d'exposició, fins les característiques ambientals, com els núvols o la boira. Els algoritmes existents avui en dia realitzen càlculs exhaustius que contemplen totes aquestes variables [SP94], amb costos computacionals elevats, que fugen de la interactivitat desitjada. En aquesta secció es mostra l'adaptació d'alguns d'aquests algoritmes, restringint-los en el seu domini per obtenir resultats interactius satisfactoris. Aquests resultats es centraran en obtenir simulacions interactives de il·luminació que permetin moviments de càmera per el interior dels models 3D generats, observant la il·luminació natural provinent del exterior, i amb la possibilitat d'editar dinàmicament les obertures, observant els canvis de il·luminació produïts.

3.1 Direcció de la llum

La simulació realista de la il·luminació contempla, entre altres molts factors, el càlcul de la il·luminació directa i indirecta. En el cas de la il·luminació natural, el càlcul de la il·luminació indirecta és molt costós, anul·lant així la possibilitat de crear un sistema de càlcul interactiu. És per aquest motiu que l'estudi s'ha realitzat considerant tan sols el càlcul de il·luminació solar directa.

Es pot considerar que el Sol és un tipus de llum omnidireccional, ja que emet raigs en qualsevol direcció. Però degut a la distància i a les dimensions de la Terra es pot aproximar com una font de llum direccional. Per tant, el càlcul de la il·luminació solar directa depèn tan sols de la direcció en que incideixen els raigs solars sobre la superfície. [PSS99] Proposa una formulació per el càlcul d'aquesta direcció basada en els següents paràmetres: longitud, latitud, dia del any, hora, minut i zona horària. El resultat son els dos angles del vector d'incidència de la llum solar sobre un punt concret de la geografia, en un dia i hora concrets. El càlcul de il·luminació directa d'una escena amb una font de llum direccional és senzill, amés d'existir avui en dia solucions ja implementades en hardware i que ofereixen una alta eficiència en processos interactius [WesGL].

3.2 Ombres

La visualització d'ombres és una característica molt útil per una correcta simulació de il·luminació. Existeixen molts algoritmes que calculen ombres en una escena [WPF90]. Per els objectius del treball present s'han analitzat aquells que permeten un càlcul interactiu, d'acord amb la simulació de il·luminació comentada en el apartat anterior.

El Ray Tracing és un algoritme de càlcul de il·luminació que inclou entre altres el càlcul d'ombres. Aquest, tot i ser un mètode ràpid i eficient, depèn de la posició actual de la càmera en l'escena, i per tant cal refer el càlcul per cada moviment d'aquesta. Conclusions similars es podrien extreure d'altres algoritmes basats en el traçat de raigs, com el Path Tracing, Light Tracing o Photon Mapping [SP94].

El Projection Shadows [Kil01] permet calcular les ombres a partir de la projecció dels polígons que les generen sobre els polígons que les han de rebre. En una escena complexa el seu cost pot

ser elevat ja que, suposant que qualsevol polígon pot generar ombres, el seu cost és quadràtic al nombre de polígons.

El Texture Shadows realitza el càlcul de totes les ombres de l'escena, generant textures per cada polígon, oferint així una alta interactivitat en els canvis de posició de la càmera. Però els resultats no disposen de molta qualitat degut als problemes inherents a la utilització de textures, com per exemple problemes d'aliasing.

Finalment, el *Stencil Shadows* [Kil01][Kil99][Kwo02] és un algoritme que basa la seva eficiència en la seva implementació sobre dispositius hardware senzills com OpenGL, i que permet un càlcul interactiu d'ombres independent a la vista actual. El Stencil Shadows parteix de la idea original del *Volume Shadows*. Aquest algoritme proposa la construcció de volums en el espai, generats per els polígons que creen les ombres. Llavors divideix l'espai en dos. Per un costat estaran els objectes que intersecten amb aquests volums, els quals queden en ombra. Per altra banda està la resta, no afectats per l'ombra (veure Fig.9). L'algoritme es pot dividir doncs en dos etapes: la construcció dels volums i el càlcul de interseccions. Però és en aquesta segona fase on l'algoritme passa a ser depenent de vista, ja que fa falta un traçador de raigs per determinar la inclusió dins dels volums calculats de cada punt de cada objecte. Per evitar això, el Shadow Volumes emprava les funcionalitats bàsiques dels dispositius gràfics hardware existents avui en dia.

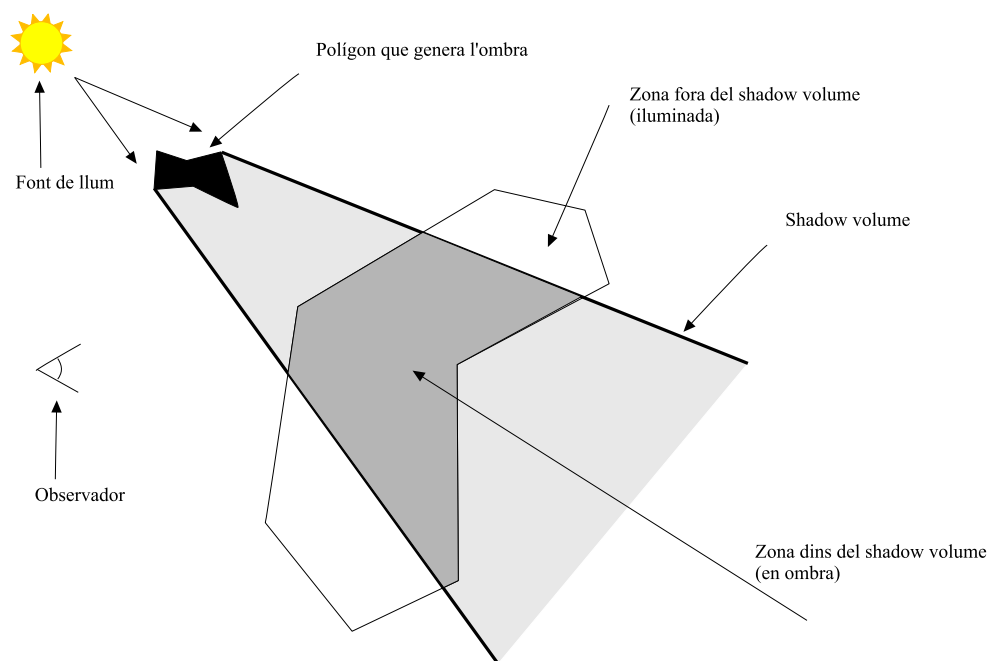


Figura 9: Vista seccional 2D del Shadow Volume.

Aquest algoritme està àmpliament documentat amb exemples en [Kil01] i [Kwo02]. El treball present s'ha basat en adaptar-lo a les necessitats d'un entorn virtual arquitectònic. A continuació es comentaran totes les aproximacions realitzades fins arribar a una situació òptima d'eficiència amb els resultats desitjats.

3.2.1 Càlcul de volums

Els volums a generar han d'estar formats a partir de les projeccions dels polígons que generen les ombres i la incidència de il·luminació solar sobre aquests. Suposant dos tipus de fonts de llum,

omnidireccional i direccional, podem obtenir dos tipus de volums: piràmides o prismes, respectivament. En la secció 3.1 es comentaven els motius per l'elecció d'una font de llum direccional. Per tant, els volums seran prismes (veure Fig.10a). La construcció d'aquests prismes es realitzarà projectant els vèrtexs dels polígons que els generen en la mateixa direcció de la il·luminació solar.

No tots els polígons de l'escena generen ombres. Per exemple, el terra de la planta no pot generar ombres. També es pot observar donada una seqüència de parets, que es pot prescindir de la meitat dels seus polígons. Aplicant un algoritme simple de visibilitat [FDFHP], i considerant el punt de vista en la font de llum, podem descartar aquells que son visibles, utilitzant els no visibles com a generadors d'ombres (veure Fig.10b).

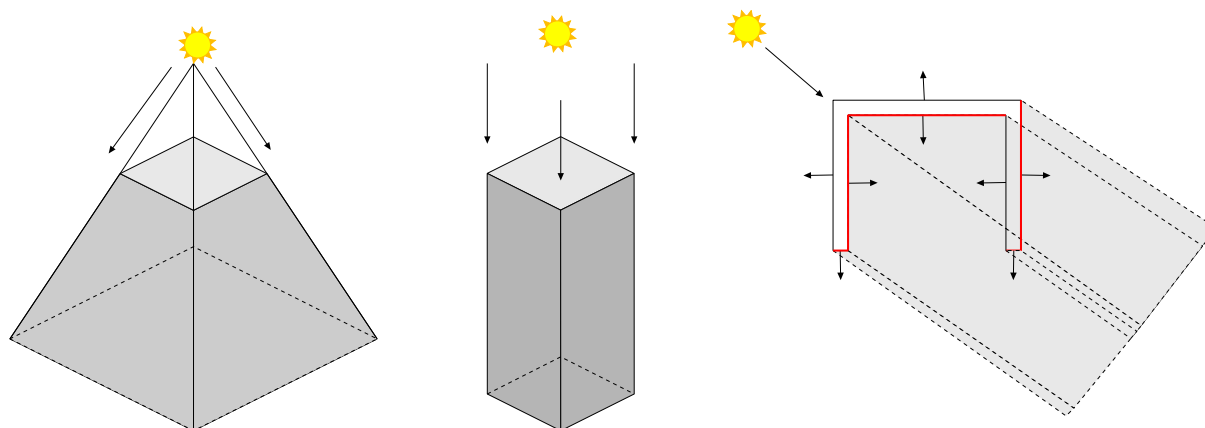


Figura 10: A l'esquerra (a) tipus de Shadow Volumes segons el tipus de font de llum. A la dreta (b) la selecció dels polígons que generen ombres.

3.2.2 Dins o fora del volum

Per determinar si un polígon o part d'ell es troba dins d'un Shadow Volume es té que tenir en compte la posició i direcció de la càmera, i comprovar píxel a píxel de la imatge que s'observa des d'aquest punt de vista. Existeixen diferents aproximacions proposades per [Kil99][Kil01] i que s'han estudiat per adaptar-les al tipus d'escena i il·luminació requerits per aquest treball.

Una primera aproximació proposa traçar raigs des del observador fins el infinit. Cada raig conté un flag binari que se li assigna 1 quan entra en un Shadow Volume, i se li assigna 0 quan surt d'un Shadow Volume (veure Fig.11a). Com que cada raig correspon a un píxel de la imatge final, tots els píxels que tinguin el flag a 1 es consideraran en ombra.

Però aquesta solució no és útil quan existeixen dos o més volums que s'intersecten, ja que es pot considerar que un raig surt d'un volum quan realment està entrant en un altre (veure Fig.11b), i aquest cas es produeix moltes vegades en un model arquitectònic, com per exemple entre les parets.

Una segona aproximació proposa la utilització d'un comptador enter. En aquest cas, cada cop que un raig entri en un volum, el comptador incrementa en un, i cada cop que surt decrementa en un (veure Fig.12a). Al final, els píxels corresponents a raigs amb comptadors diferents a 0 estaran en ombra. Aquest algoritme també rep el nom de *Z-pass*.

Però aquest mètode no contempla la possibilitat de situar la càmera en el interior de un Shadow Volume, ja que el comptador no s'inicialitza correctament (veure Fig.12b). Si situem la càmera en

el interior de la planta per observar la il·luminació exterior segurament es donarà aquest cas.

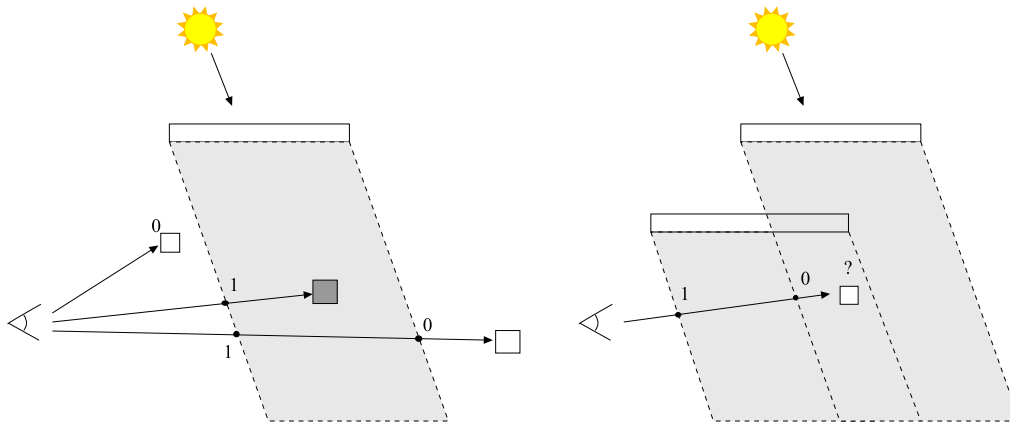


Figura 11: Primera aproximació del algoritme. A l'esquerra (a) es mostra un exemple. A la dreta (b) es mostra el problema d'aquesta aproximació.

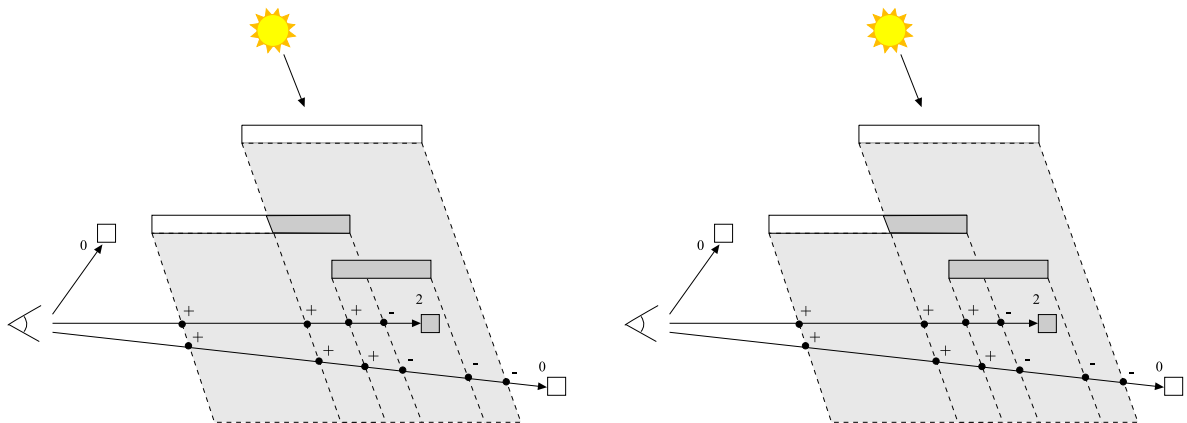


Figura 12: Aproximació Z-pass. A l'esquerra (a) es mostra un exemple de funcionament. A la dreta (b) es mostra el problema d'aquesta aproximació

Per solucionar el problema del Z-pass existeix el *Z-fail*. Aquest algoritme proposa traçar raigs des del infinit fins el observador, emprant el mateix mètode del comptador. Així la posició de la càmera és independent, ja que els comptadors sempre estan inicialitzats correctament en el infinit (veure Fig.13).

3.2.3 Límits dels volums

Suposem que l'observador es situa dins un Shadow Volume i la direcció d'observació és la mateixa que la direcció d'expansió del volum. Fins ara no s'ha determinat un límit real per aquests volums, però en aquest cas és necessari definir una tapa en un extrem final, ja que en cas contrari el algoritme Z-fail no visualitzaria correctament l'ombra (veure Fig.14a). Però aquest límit no pot ser petit, ja que l'ombra es te que estendre per tota l'escena si és necessari. Alhora tampoc pot ser superior al viewport màxim de l'escena, ja que el sistema depèn fortament del sistema de visualització (veure Fig.14b). Així doncs, una bona mesura és establir un viewport suficientment gran i definir les tapes a una distància lleugerament inferior als límits d'aquest viewport [EK02].

Si l'observador es troba en el interior d'un Shadow Volume i la seva direcció de visualització és l'oposada a la direcció d'expansió del volum, es pot produir un efecte semblant al anterior, ja que

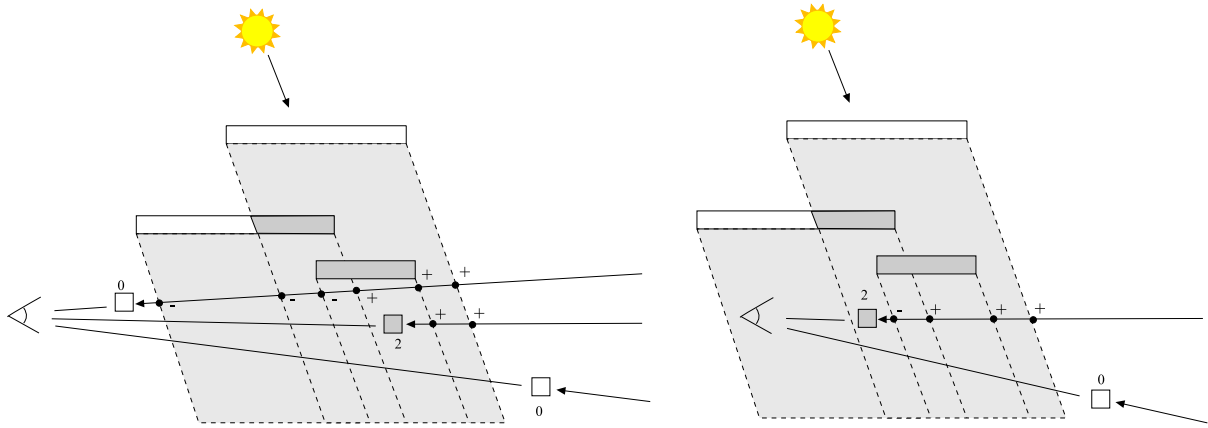


Figura 13: Aproximació Z-fail amb la càmera fora i dins dels shadow volumes.

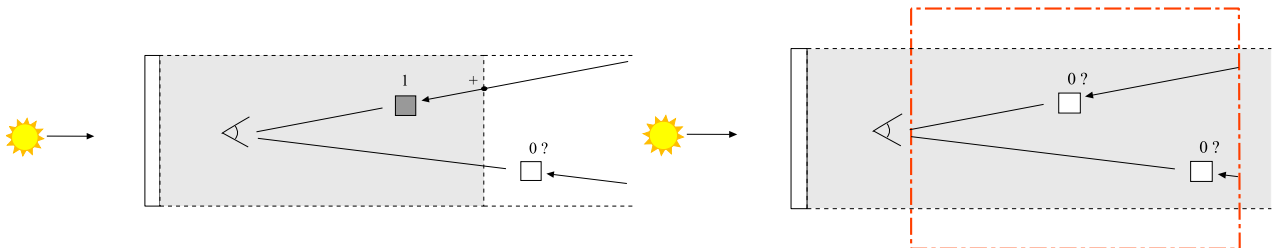


Figura 14: A l'esquerra (a) es mostra la necessitat d'expandir el volum fins al infinit. A la dreta (b) es mostra per contra la necessitat de que aquest límit no sigui superior als límits del viewport (marcat en vermell i línia puntejada)

és necessària una tapa que tanqui el volum per el seu extrem superior, des del polígon que genera l'ombra.

Amés de la necessitat d'establir límits als volums, també existeix la possibilitat de que es produeixin overlappings entre polígons en dos casos: un polígon de la planta coincideix de forma co-planar amb un dels laterals d'un Shadow Volume, i quan la tapa superior del volum queda sobreposada amb el polígon que la genera. El primer cas es soluciona a nivell de visualització, aplicant un petit desplaçament visual fins a davant o fins darrera, en funció de si es decideix que aquell polígon esta en ombra o no. El segon cas és idèntic al primer, amb la particularitat que el desplaçament té que ser de forma que el polígon quedi en el interior del Shadow Volume.

3.2.4 Implementació en OpenGL

Tot i que usualment el rendiment d'un algoritme no depèn directament de la implementació realitzada, en aquest cas és molt important la utilització d'una llibreria gràfica com OpenGL. L'algoritme Stencil Shadows n'és un bon exemple. Observant el mètode original del Volume Shadows i les seves variacions es pot arribar a la conclusió que es tracta d'un sistema dependent de vista i amb un cert cost computacional que fan d'ell un mètode no interactiu. Les implementacions realitzades en el Stencil Shadows [Kil99][Kil01] aprofiten característiques bàsiques dels dispositius gràfics hardware convencionals avui en dia, convertint-lo en un sistema interactiu i de baix cost. A continuació es mostren els tres passos que componen aquesta implementació, la qual es pot prendre en sí mateixa com un algoritme independent. Abans de procedir a la redacció d'aquests passos és convenient que el lector consulti nocions bàsiques de OpenGL [WesGL] en el referent al treball amb el Depth

Buffer, Stencil Buffer i el pipeline general d'execució.

La primera fase inicialitza els buffers i renderitza l'escena amb la il·luminació solar activada.

```
ActivarIluminacioSolar();

glEnable(GL_DEPTH_TEST);           // Activa i configura el Depth test
glDepthFunc(GL_LEQUAL);
glDepthMask(1);

glDisable(GL_STENCIL_TEST);        // Desactiva el Stencil Test
glClearStencil(0);

glColorMask(1,1,1,1);              // Inicialitza el buffer de color
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT |
        GL_STENCIL_BUFFER_BIT);

glEnable(GL_CULL_FACE);
PintarEscena();                    // Pinta tota la escena
```

La segona fase desactiva la il·luminació solar i evita l'escriptura sobre el Depth Buffer. Alhora implementa el algoritme Z-fail, configurant el Stencil Buffer de forma que realitzi la tasca de comptador per a cada raig traçat corresponent a un píxel. Una vegada configurats aquests paràmetres es calculen tots els Shadow Volumes de l'escena i es dibuixen, primer les cares posteriors i després les frontals, modificant així els valors dels comptadors assignats a cada píxel en el Stencil Buffer, incrementant en primer lloc, i decrementant en segon lloc. Es té que deixar clar que el dibuixat tan sols es produeix a nivell intern sense arribar al buffer de color, i per tant, sense dibuixar en pantalla. Tot i que no es dibuixa en pantalla, és necessari que aquestes cares passin el Depth Test associat per evitar treballar amb volums que es trobin per darrera del volum real a comprovar. És per aquest motiu que el Depth Buffer es bloqueja, però no el Depth Test, permetent així treballar els dos sistemes per separat.

```

DesactivaTotaIluminacio();

glEnable(GL_DEPTH_TEST);                // Activa el Depth Test
glDepthFunc(GL_LESS);                   // Desactiva l'escriptura del Depth Buffer
glDepthMask(0);

glEnable(GL_STENCIL_TEST);               // Activa el Stencil Test
glStencilFunc(GL_ALWAYS, 0, 0);          // Configura el Stencil Test de forma
                                        // que sempre el passi

glStencilMask(~0u);                      // Activa l'escriptura del Stencil Buffer

glColorMask(0,0,0,0);                   // Desactiva l'escriptura del Color Buffer

glEnable(GL_CULL_FACE);                  // Activa el pintat de cares frontals o posteriors

glStencilOp(GL_KEEP, GL_INCR, GL_KEEP);  // Configura el Stencil Test per la segona comprovació de forma que
                                        // incrementi en 1 el valor del buffer si no passa el Depth Test i passa el
                                        // Stencil Test (el passa sempre)

glCullFace(GL_FRONT);                   // Pinta només las cares frontals dels
PintaShadowVolumes();                   // shadow volumes

glStencilOp(GL_KEEP, GL_DECR, GL_KEEP);  // Configura el Stencil Test per la segona comprovació de forma que
                                        // decrementi en 1 el valor del buffer si no passa el Depth Test i passa el
                                        // Stencil Test (el passa sempre)

glCullFace(GL_BACK);                    // Pinta només las cares posteriors dels
PintaShadowVolumes();                   // shadow volumes

```

En la tercera fase es renderitza una altra vegada l'escena, però tan sols s'actualitzen els píxels que estiguin en ombra, o sigui, aquells el valor dels quals en el Stencil Buffer sigui diferent a 0. En aquest cas la il·luminació solar resta desactivada, i per tant aquests píxels tan sols es veuran il·luminats per la resta de fonts de llum de l'escena, que en el nostre cas no hi ha cap font més de llum.

```

glEnable(GL_LIGHTING);                   // S'activa la il·luminació, però es manté
                                        // desactivada la llum solar

glEnable(GL_DEPTH_TEST);                 // S'activa el Depth Test, però es desactiva
glDepthFunc(GL_EQUAL);                   // l'escriptura del Depth Buffer, utilitzant
glDepthMask(0);                           // els valors del test anterior

glEnable(GL_STENCIL_TEST);               // S'activa el Stencil Test
glStencilFunc(GL_NOTEQUAL, 0x0, ~0u);    // Es configura per a que només passin els valors
                                        // diferents a 0

glStencilOp(GL_KEEP, GL_KEEP, GL_KEEP);  // No es realitza cap operació

glColorMask(1,1,1,1);                   // S'activa el Color Buffer
PintarEscena();                           // Pinta tota la escena

```

S'han aplicat algunes millores al algoritme, com el ús eficient de CallLists per agilitzar l'accés a la geometria a renderitzar, així com el càlcul dels Shadow Volums. També s'ha fet ús del Blending, cosa que ha permès barrejar el negre absolut en les zones en ombra amb un valor de il·luminació ambient, aplicat tan sols sobre aquestes zones. Aquesta característica permet observar millor la geometria en les zones en ombra.

4 Resultats

Els resultats d'aquest estudi es poden localitzar en dos àrees: la generació de models 3D arquitectònics i la simulació de il·luminació natural en aquests.

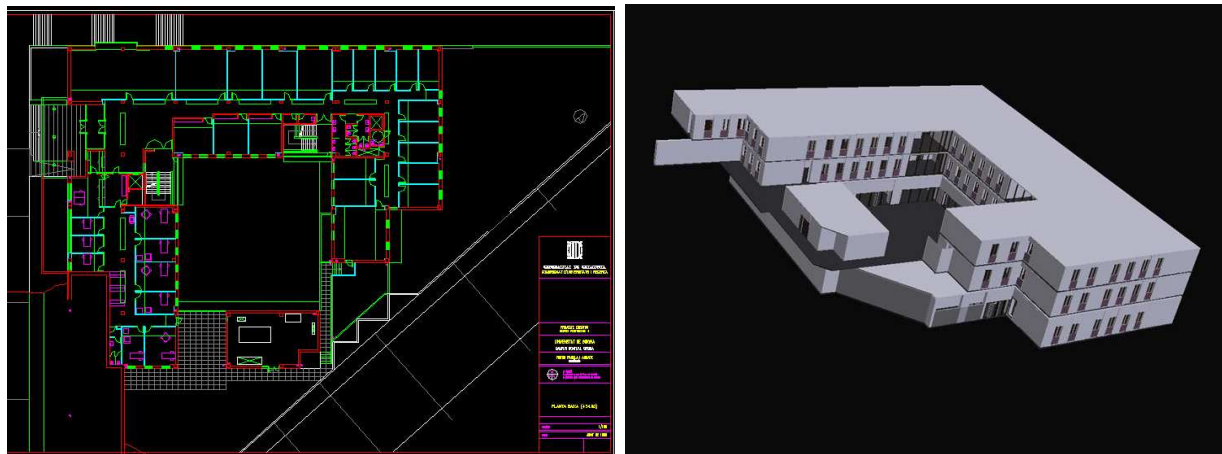


Figura 15: A l'esquerra es mostra el model 2D del edifici P4 de la Universitat de Girona. A la dreta es mostra el model 3D generat.

La generació de models 3D arquitectònics a partir dels seus respectius models 2D en planta permet obtenir estructures vàlides que mostren la distribució d'espais general, tot i no entrar en detalls molt precisos i que queden fora del traçat general del plànol en planta.

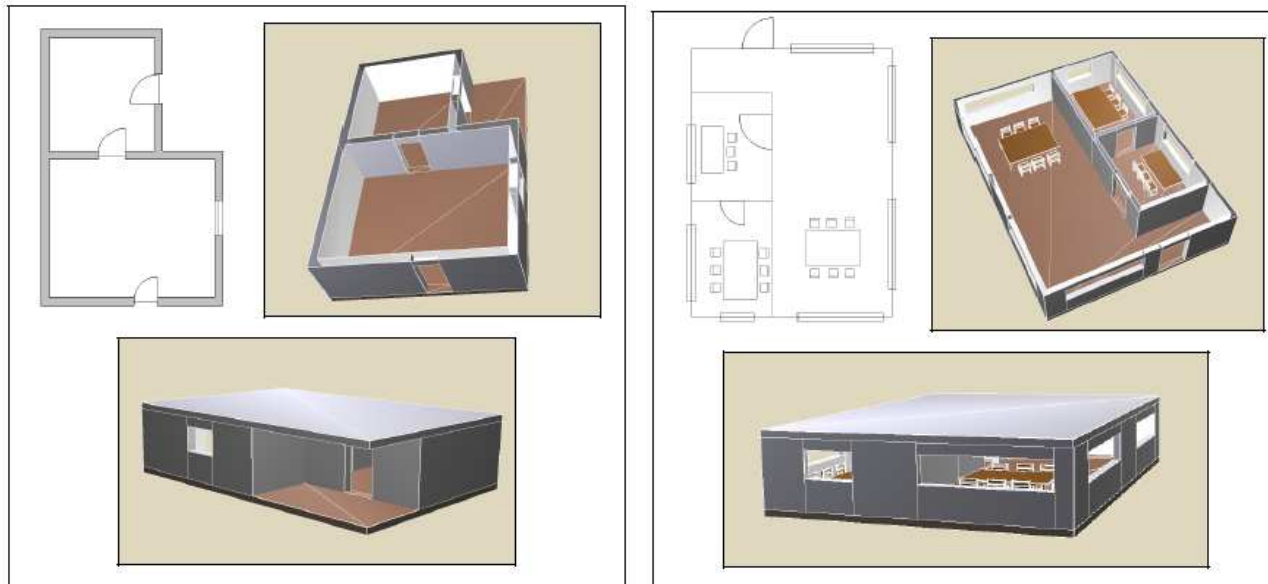


Figura 16: Exemples de generació de models 3D a partir de models 2D.

La simulació de il·luminació natural permet aproximar la situació real d'un element arquitectònic en el seu entorn. Aquest realisme s'ha centrat en dos aspectes: incidència de la llum solar i generació d'ombres en el interior de la planta. Tot i no tractar-se de simulacions de il·luminació realista, la possibilitat de modificar la geometria i les ombres que aquesta genera de forma interactiva, permet al dissenyador observar com varia la il·luminació en el interior de la planta en funció del dia, hora i situació geogràfica.

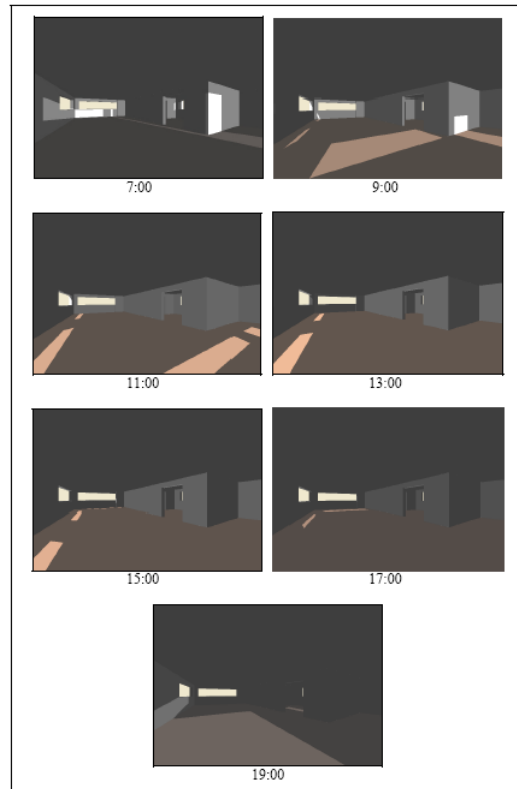
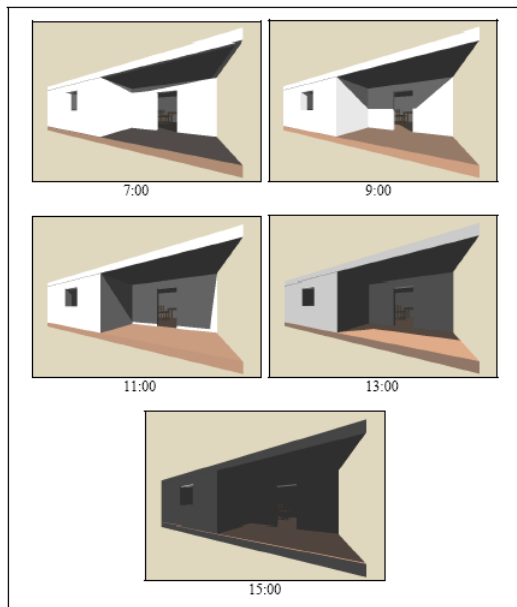


Figura 17: Resultats dels càlculs interactius per a la simulació de il·luminació solar a diferents hores del dia.

5 Treball futur

El desenllaç d'aquest treball ha generat aportacions amb possibles modificacions i ampliacions. Algunes d'elles responen a solucions a problemes no tancats. Altres es refereixen a noves característiques que permeten agregar un valor afegit al resultat.

En la generació de models 3D han quedat oberts dos problemes: cerca de parets en elements estructurals no continus, deguts per exemple a unions errònies entre segments d'una polilínia per manca de precisió, i la millora en el ajustament dels terres i sostres al perfil real de la planta, amb perfils còncaus per exemple. Nous elements interessants a afegir són els sostres de diferents alçades i inclinacions, escales i obertures entre plantes.

En la simulació de il·luminació natural, s'han detectat certs problemes visuals derivats del algoritme emprat, i donats en casos molt concrets on és molt difícil discernir entre límits d'ombres quan els seus polígons són co-planars entre ells o entre la geometria de la planta. Seria necessari analitzar diferents casos pràctics i establir una nova variant del algoritme que permeti resoldre'ls. Com a nous elements interessants a afegir són la simulació del cel i la il·luminació indirecta que prové d'ell, i l'aplicació de la cromacitat del color de la llum solar en funció de la hora del dia simulada.

Referències

- [Arc01] Autodesk Architectural Studio. Getting Started and New Features Guide. Autodesk, 2001.
- [Aut97] AutoCAD Release 14. Manual de usuario. Autodesk Development, 1997.
- [AuxDXF] DXF Reference, v. U14.1.04. Autodesk Development
- [EK02] Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering *Cass Everitt and Mark J.Kilgard*. NVIDIA Corporation, 2002.
- [FDFHP] Introduction to Computer Graphics. *James D.Foley, Andries van Dam, Steven K.Feiner, John F. Hughes, Richard L.Phillips*. Ed. Addison Wesley.
- [Kil01] Robust Stencil Shadow Volumes. *Mark J.Kilgard*. NVIDIA Corporation. CEDEC 2001, Tokyo, September 4, 2001.
- [Kil99] Improving Shadows and Reflections via Stencil Buffer. *Mark J.Kilgard*. NVIDIA Corporation. Game Developer Conference, March 16, 1999, pp. 204-253.
- [Kwo02] The Theory of Stencil Shadow Volumes. *Hun Yen Kwoon*. GameDev.com, 2002.
- [PSS99] A Practical Analytic Model for Daylight. *A.J. Preetham, Peter Shirley, Brian Smits*. Research Report, University of Utah.
- [SP94] Radiosity Global Illumination. *François X.Sillion, Claude Puech*. Morgan Kaufman Publishers, 1994.
- [WesGL] OpenGL Programing Guide. Ed. Adisson-Wesley
- [WPF90] A Survey of Shadow Algorithms. *Andrew Woo, Pierre Poulin, Alain Fournier*. IEEE Computer Graphics and Applications, November 1990, pp. 13-32.