



ENTORN DE DESENVOLUPAMENT DE REALITAT AUGMENTADA

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Ferran Poveda Abansés
i dirigit per
Enric Martí Godia
Bellaterra, Juny de 2008

El sotasignat, Enric Martí Godia

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per Ferran Poveda Abansés

I per tal que consti firma la present.

Signat:

Bellaterra, Juny de 2008

*“ The Book: Curiously the only
thing that went through the
mind of the bowl of petunias, as
it fell, was, 'Oh no, not again' ”*

*The Hitchhiker's Guide to the Galaxy
Douglas Adams*

Agraïments

En primer lloc agrair la seva col·laboració i recolzament al meu director de projecte, Enric Martí. Sense ell, aquest projecte no hagués estat possible.

També voldria agrair la paciència de tots aquells que m'han acompanyat dia a dia, que m'aconsellen quan ni jo m'entenc i que m'han ajudat a arribar on sóc. Especialment, a la persona que ho ha fet més que ningú en els darrers 5 anys i també en aquest projecte, l'Helena.

Finalment, gràcies de tot cor als meus pares, Francesc i Azucena, els quals han estat una gran companyia, guia i ajuda per afrontar moltes coses en aquesta vida, també en la meva carrera i en aquest projecte.

Índex

Capítol 1: Introducció	1
1.1 Què és la Realitat Augmentada?	1
1.2 Quina és la seva aplicació?	3
1.3 Motivació: per què fer aquest projecte?	4
1.4 Objectius del projecte	5
1.5 Planificació temporal del projecte	6
Capítol 2: Antecedents	7
2.1 Elements que intervenen en un entorn de Realitat Augmentada	7
2.1.1 Dispositius de visualització	7
a) Les ulleres màgiques: HMD	8
b) Les finestres cap al món augmentat: Handhelds	9
c) Un clàssic: el monitor	10
d) Projecció: elements virtuals literalment al món real	10
2.1.2 Obtenció i tractament de dades de l'entorn	10
a) Sensors	11
b) Visió per computador	11
c) Sistemes mixtos	11
2.1.3 Interfície amb l'usuari	12
2.2 On s'emmarca aquest projecte?	13
Capítol 3: Anàlisi, disseny i implementació	15
3.1 Estudi de viabilitat	15
3.2 Anàlisi de requeriments	16
3.3 Disseny de l'entorn	18
3.3.1 Captura d'imatges	19

3.3.2	Mescla d'objectes virtuals i realitat	19
3.3.3	Accés al flux de vídeo	21
3.4	Implementació de l'entorn	21
3.4.1	Interfície i presentació	21
3.4.2	Captura d'imatges	21
3.4.3	Gestió del flux de vídeo i mescla d'objectes virtuals i realitat	23
a)	Pas del flux de captura a l'espai tridimensional	23
b)	Gestió del flux de vídeo	24
c)	Superposició d'elements virtuals al flux de vídeo	24
d)	Construcció de l'escena augmentada	25
3.5	Disseny de la solució d'RA	26
3.5.1	Seguiment	26
3.5.2	Alineació	28
3.6	Implementació de la solució d'RA	29
3.6.1	Seguiment	29
a)	Detecció: Pretractament de les imatges	29
b)	Detecció: Localització de patrons a l'escena	30
c)	Identificació del patró	31
3.6.2	Estimació de posició i alineació	33
Capítol 4: Anàlisi de resultats		37
4.1	Resultats de l'entorn	37
4.2	Resultats de la solució	38
a)	Rendiment de l'augmentació	38
b)	Robustesa de l'augmentació	39
4.3	Resultats del disseny i la implementació	42
Capítol 5: Conclusions		45
5.1	Visió general del projecte	45
5.2	Millores, ampliacions i futur del projecte	46
5.3	Valoració personal	48

Índex de figures

1. Readaptació de l'esquema del " <i>virtuality continuum</i> " de Milgram i Kishino	2
2. Realitat augmentada vs <i>chiron</i>	3
3. Exemples d'aplicació de la Realitat Augmentada (en l'ús per l'obtenció de biòpsies, en la navegació <i>GPS</i> i en un videojoc)	4
4. Diagrama Gannt de planificació del projecte	6
5. Esquema de funcionament de vídeo i <i>see-through HMDs</i>	8
6. Realitat Augmentada a Handheld	9
7. Realitat Augmentada al monitor "clàssic"	10
8. Seguiment explícit d'un marcador i seguiment per punts característics de l'escena	11
9. Guants sensors de gestos amb interfície <i>hàptica</i>	12
10. Esquema del flux de treball d'un entorn genèric de Realitat Augmentada basat en Visió per Computador	16
11. Flux adaptat per l'entorn de proves	18
12. Esquema del disseny de l'entorn i l'aplicació d'interfície	19
13. Representació esquemàtica dels models "centrat en la càmera" i "centrat en l'objecte"	20
14. Graf implementat amb <i>DirectShow</i> per a la captura d'imatges des de la <i>webcam</i>	22
15. Gestió del flux de vídeo	24
16. Modes de superposició dels models virtuals	25
17. Etapes del processament de vídeo per la solució d'RA	26
18. Patró marcador	27
19. Exemples de captura, detecció, identificació, estimació de postura i alineació	28
20. Flux de la detecció del patró	31

21. Homografia de la imatge capturada per reconstruir el seu contingut	32
22. Procés d'identificació del patró	33
23. Reconstrucció de la matriu de visualització 3D a partir de l'homografia	34
24. Model de perspectiva dèbil	35
25. Generació de la matriu de visualització per a OpenGL	36
26. Generació de la matriu de projecció per a OpenGL	36
27. Mostres de captures utilitzades en els tests	38
28. Muntatge de la captura d'imatges per les proves	39
29. Muntatge de la captura d'imatges per la prova de detecció i identificació en un entorn heterogeni	40
30. Mostres de les captures utilitzades en els tests de distància de detecció	41
31: Muntatge de la captura d'imatges per les proves d'angle	42

Capítol 1

Introducció

El primer dispositiu de visualització gràfica en un computador data de l'any 1950. Es tractava d'una pantalla de tecnologia CRT que formava part del computador *Whirlwind*, del MIT (*Massachusetts Institute of Technology*). En aquest moment es pot dir que neix la primera interfície gràfica amb l'ordinador.

Uns quants anys més endavant, al 1963, el famós investigador Ivan Edward Sutherland [1], presentava l'anomenat *SketchPad*, un software d'interfície amb el qual es definia un CAD (*Computer Aided Design*; una eina de disseny gràfic assistida per ordinador) que permetia l'edició directa en pantalla mitjançant un llapis òptic. De fet, el llapis òptic també era novetat; l'invent s'atribueix al propi germà d'Ivan.

Poc després, el mateix Sutherland plantejava arribar molt més enllà en la forma de presentar dades en un *display*; al seu article *The Ultimate Display* [2] presenta una primera idea revolucionària: obrir les portes a la possibilitat de recrear en un *display* imatges que no tinguin perquè respondre a la lògica del món real. Això, com ell explica, podria portar-nos *al País de les Meravelles d'Àlicia*. Es pot dir que és el primer en exposar una idea sobre Entorns Virtuals (coneguts també com a Realitat Virtual) i sobre una altra tecnologia, que considera la possibilitat d'afegir objectes sintètics al món real que veiem amb els nostres ulls: **La Realitat Augmentada**.

1.1 Què és la Realitat Augmentada?

Ivan Sutherland va continuar treballant en aquesta línia amb els primers dispositius de visualització *HMD* (*Head Mounted Display*) amb els quals, en les seves investigacions properes al 1968, va poder presentar un quadre de malla superposat al món real. Aquí es quan es pot dir que neix definitivament la Realitat Augmentada (a partir d'ara la notarem també com a RA). Aquesta tecnologia va ser investigada a fons en els següents anys, però no va establir-se definitivament fins a principis dels anys 90, quan es van aconseguir millores en algorítmica i la potència de còmput havia augmentat, permetent així uns resultats més robustos.

Sovint el terme RA es confon amb el terme Realitat Mixta. Per això Paul Milgram i Fumio Kishino defineixen, l'any 1994 en el seu article *A Taxonomy of Mixed Reality Displays* [3], l'anomenat *virtuality continuum* amb el qual es precisen on comencen i on acaben aquestes tecnologies. A partir d'aleshores, la definició del *virtuality continuum* ha estat presa com estàndard *de facto* per la majoria d'investigadors.

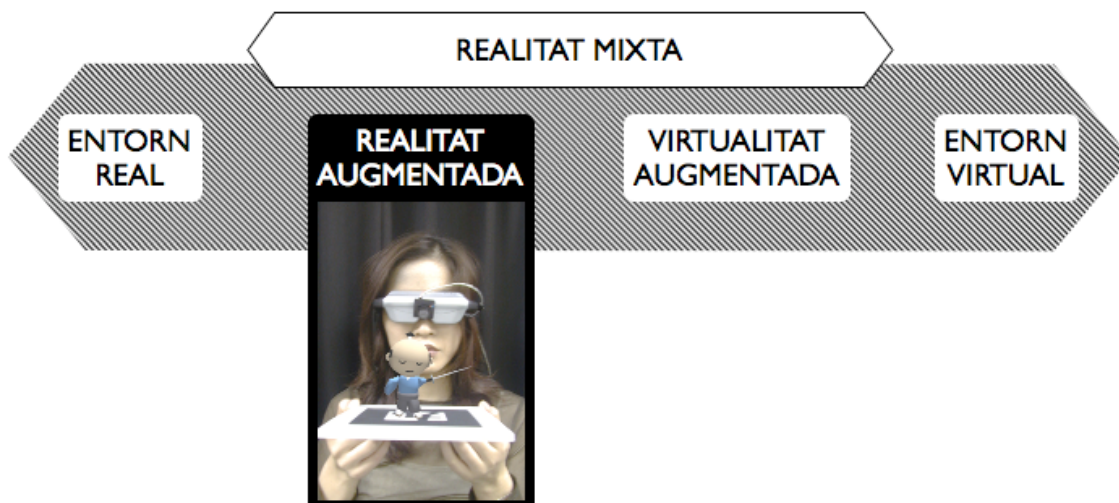


Figura 1: Readaptació de l'esquema del "virtuality continuum" de Milgram i Kishino

La figura 1 mostra el plantejament de Milgram i Kishino agrupant de dreta a esquerra les diferents possibilitats, que van des de l'entorn purament real al purament virtual. La Realitat Augmentada s'inclou doncs, com una part de la Realitat Mixta juntament amb la Virtualitat Augmentada. Ambdues tenen en comú la fusió d'elements reals i virtuals en diferent grau. Segons aquest grau, i sense ser estrictes, podem definir que la Realitat Augmentada és la integració d'objectes virtuals sobre el món real i la Virtualitat Augmentada esdevé quan afegim objectes del món real a un Entorn Virtual. Aquesta definició, tot i aproximar molt bé aquests conceptes, no és del tot correcta. Cal afinar-la més:

L'investigador en Realitat Augmentada, Ronald T. Azuma, en els seus reculls *A Survey of Augmented Reality* [4] i *Recent Advances in Augmented Reality* [5] defineix la RA esmentant les característiques pròpies de la tecnologia, construint així una definició més precisa. Azuma defineix l'existència de Realitat Augmentada sempre que:

- es presentin elements virtuals en un entorn real,
- existeixi una relació en temps real entre l'element virtual i la realitat, i
- que, aquest element, estigui alineat en l'espai de forma tridimensional.

El pas dels anys podria haver afectat a aquesta definició, que data del 1997, ja que el fet de la tridimensionalitat d'alineació pot haver deixat de ser necessària. Existeixen aplicacions com ara la recent proposta pel *Google Android Developer Challenge: Enkin* [6] que, tot i no utilitzar una relació purament 3D entre l'element virtual i la realitat, sí és considerada Realitat Augmentada. El que sí segueix definint en essència la Realitat Augmentada és l'alineació de l'element virtual amb la realitat, com mostra la figura 2a. Si l'element virtual perd aquesta propietat no estariem parlant d'RA sinó d'una simple superposició (com ara un *chyron* amb notícies afegides en un programa de televisió; figura 2b).



(a)

(b)

Figura 2: Realitat augmentada vs chyron

1.2 Quina és la seva aplicació?

En el camp de la Realitat Augmentada podem trobar moltes investigacions dedicades als algoritmes implementats per aconseguir resultats robustos i adaptats a tot tipus de casuístiques. Però també s'han realitzat moltes d'altres dirigides a utilitzar aquesta tecnologia com a interfície per a finalitats molt diverses. Parlem d'usos que van des de l'aplicació en medicina fins a àmbits militars. També s'ha utilitzat en interfícies que permeten fer anotació i treballar més còmodament a l'hora de fer reparacions i productes manufacturats. A més a més, pot ser utilitzada com a mitjà de visualització per l'ensenyament o el disseny com un CAD. I és un dels camps considerats per la investigació com a interfície d'entreteniment des del més pur *gaming* a la seva integració en plataformes multimèdia per a llars. Veiem-ne alguns exemples.

En entorns **mèdics** aquesta eina ha estat considerada per millorar tècniques d'operació que minimitzin la invasió del pacient. Com proposaven l'any 1996 Andrei State i els seus companys de la Universitat de Nord-Califòrnia a l'article [7], en una operació laparoscòpica habitualment es fa ús de microcàmeres introduïdes en el cos del pacient per treballar, però mitjançant RA es pot millorar aquesta interfície superposant imatges obtingudes per ultrasons sobre pantalles o el propi cos del pacient amb l'objectiu de facilitar la navegació del cirurgià (figura 3a). Altres propostes han estat orientades cap a l'aprenentatge [8] o fins i tot per permetre observar de forma virtual el fetus d'una dona embarassada superposat a la seva panxa [9].

Un altre dels camps majoritaris de les aplicacions de l'RA es troba en l'**anotació i visualització**: en són grans exponents les aplicacions de navegació com la recent presentada *Enkin* [6] (figura 3b), o bé altres aproximacions com *MARA* del *Nokia Research Center* [10]. També s'han ideat entorns per ajudar a la manufactura o reparació de tot tipus de mecanismes; un bon exemple és el de les recents investigacions del *Fraunhofer Institute Graphische Datenverarbeitung* en la guia interactiva de reparació per la firma d'automòbils *BMW* [11].



(a) (b) (c)
 Figura: Exemples d'aplicació de la Realitat Augmentada
 (en l'ús per l'obtenció de biòpsies, en la navegació GPS i en un videojoc)

Per acabar, com ja hem destacat, l'RA també ha entrat de ple en la investigació per a l'**entreteniment**: aplicacions fins i tot ja comercials de *T-Inmersion™*, com la nova atracció del parc d'atraccions *Futurscope* [12] o bé el joc "*Eye of Judgement*" [13] de *Sony™* per a *Playstation 3* (figura 3c).

1.3 Motivació: per què aquest projecte?

La Realitat Augmentada es va començar a investigar alhora que la Realitat Virtual, però només aquesta segona ha passat a ser coneguda pel gran públic. La Realitat Virtual ha estat utilitzada en aplicacions industrials, també en parcs d'atraccions i citada en pel·lícules, a més a més ha estat focus d'experimentació en medicina pel tractament de trastorns com ara la claustrofòbia o l'agorafòbia. En canvi, la Realitat Augmentada ha estat menys difosa, però no per això menys investigada.

Limitacions tècniques en diversos sentits: *displays*, eines i còmput; i en la depuració de tots ells, han afectat a les possibilitats que ambdós sistemes podien oferir. Amb el pas dels temps (estem parlant de gairebé mig segle després del seu naixement conceptual) aquestes limitacions estan minvant, de manera que la Realitat Augmentada podria deixar de ser, en pocs anys, un mer espectacle de fira i esdevenir una tecnologia d'ús domèstic, industrial i comercial habitual. De fet, en l'actualitat la majoria de les aplicacions encara estan en una fase conceptual, no estan preparades per esdevenir projectes comercials reals. Encara hi ha molta feina a fer.

A nivell personal, fer aquest projecte respon al meu interès personal d'unir-me al desenvolupament d'una tecnologia que crec potent i amb futur. La Realitat Augmentada és un àmbit en plena efervescència en el que queda molt per fer, tant en la seva vessant més arquitectural com en les seves aplicacions. Calen algorismes que la facin més estable pel seu ús "comercial", a més a més d'aconseguir millors resultats de rendiment (per fer-la, per exemple, adaptable a dispositius mòbils de menor potència) o bé una representació més realista i integrada amb l'entorn real. Pel que fa a la seva aplicació, obre un ample ventall de solucions al que pot adaptar-se com una de les interfícies home-ordinador més completes conegudes fins a dia d'avui. Finalment, aquest camp combina els àmbits de la informàtica que més m'interessen, com són els gràfics per computador, el tractament d'imatges i la visió per computador.

1.4 Objectius del projecte

El punt de partida per plantejar aquest projecte és la intenció de crear un **entorn de proves de Realitat Augmentada**. Es tracta d'un entorn de proves pensat pel desenvolupament de nous algoritmes aplicats a aquesta tecnologia. Com veurem a l'apartat d'antecedents de la memòria, tot i portar un llarg recorregut en la investigació en múltiples línies a múltiples centres de recerca, les solucions no solen ser òptimes i sempre queda molt per aprendre i millorar.

El projecte també té per objectiu ser un entorn amb la possibilitat d'implementar proves funcionals d'aplicacions i interfícies per fer tests de concepte, sense la pretensió inicial de generar una llibreria de desenvolupament. En aquest camp ja hi ha llibreries com *ArtoolKit* [14] (i variants que han sortit per adaptar-la a diversos entorns) i *ARtag* [15]. Tot i així, tampoc seria del tot descabellat fer aquest plantejament si es poden desenvolupar nous algoritmes que no cobreixin les altres llibreries disponibles. Sobretot, tenint en compte que aquestes llibreries (almenys fins a dia d'avui) proposen una solució més tancada del que pot oferir la RA.

Per tal d'aconseguir la nostra fita seguirem els següents passos:

1. De forma introductòria hem de veure quin és l'estat de l'art de la Realitat Augmentada. Aquest però, ha de ser un aprofundiment en aquesta tecnologia i ha de servir per disposar d'un bon criteri per elaborar el projecte. Això també assegurarà que el projecte es dugui a terme de forma satisfactòria. D'això ens encarregarem al llarg del **segon capítol**.
2. El segon objectiu és analitzar totes les parts del problema, estudiar-ne a més a més els requeriments per poder dissenyar l'entorn de desenvolupament. Un cop completat, desenvolupar-lo. D'aquest procés, de les decisions preses i dels resultats obtinguts, en farem la descripció al **tercer capítol**.
3. Un cop disposem de l'entorn preparat, en farem una anàlisi de les seves capacitats i rendiment. L'objectiu és veure en profunditat quin és el comportament de l'entorn en vers l'esperat. Aquesta anàlisi la veurem al **quart capítol**.
4. Finalment, al **cinquè capítol**, després d'haver superat totes les etapes del projecte, disposarem de dades suficients per veure quins són els punts forts i febles. Aquesta anàlisi ens permetrà ser crítics amb el resultat obtingut i fer-ne una valoració detallada, a més de proposar quin pot ser el futur del projecte.

1.5 Planificació temporal del projecte

Per tal d'assegurar que el projecte serà completat satisfactòriament, i abans d'endinsar-nos en el context del projecte vam decidir fer una planificació temporal que ens ajudés a fer-ne el seguiment. Aquesta anàlisi va estar realitzada per tal de marcar les fites principals que s'havien d'assolir i els períodes que marcava cada apartat. Està definit en **Anàlisi del context, Disseny, Codificació, Anàlisi de resultats i Memòria**. L'apartat que vam suposar que requeriria més temps és, lògicament, el de la codificació. Intervenien molts factors que indueixen a problemes en el codi, laboriosos de solucionar. També poden sorgir modificacions en el disseny i les anàlisis diverses deguts a aquests canvis. Els altres apartats són més estables i, per tant, queden més ben acotats en temps. Aquesta previsió general, la durada de les tasques i la seva ordenació es resumeixen en el *diagrama de Gantt* mostrat a la figura 4.

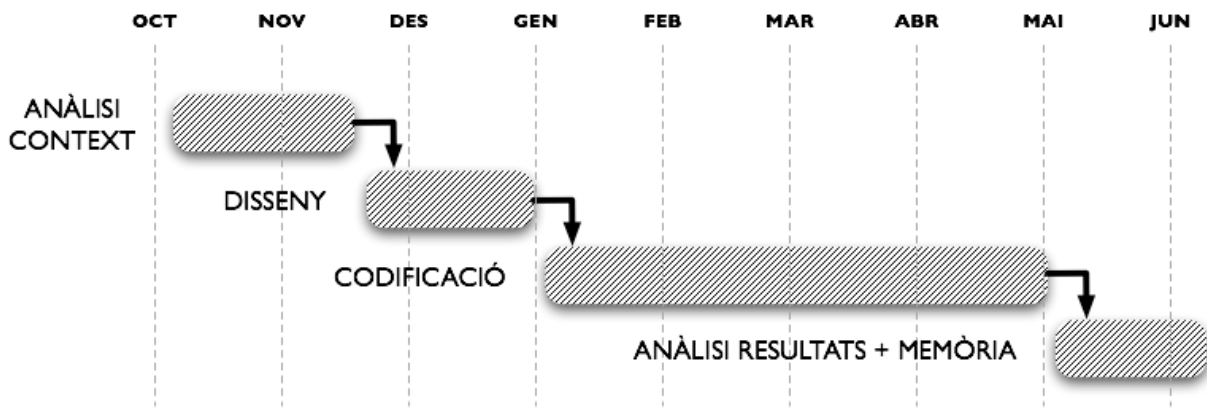


Figura 4: Diagrama Gantt de planificació del projecte

Capítol 2

Antecedents

Per tal de poder començar aquest projecte ens veiem obligats a analitzar quins han estat els treballs realitzats al llarg dels anys d'investigació en Realitat Augmentada. Això ens ajudarà a introduir-nos en aquest món i també a elaborar un criteri ferm per poder prendre decisions al llarg del projecte, tant en l'etapa de disseny com en la implementació d'aquest. Aquest capítol està dedicat a resumir els aspectes claus que poden ajudar-nos a entendre com treballar amb l'RA.

2.1. Elements que intervenen en un entorn de Realitat Augmentada

Ens plantejem construir un entorn de proves; per això hem de valorar tots els possibles detalls abans de començar. Primer de tot, és important que coneguem els diferents espais de treball d'aquesta tecnologia per així adaptar-nos correctament als diversos requeriments que aquests implicaran. Aquest apartat està basat en els *surveys* de Ronald Azuma [4][5] i en altres investigacions i aplicacions que han sortit als darrers anys i comentarem al llarg de la memòria. Els elements principals que poden estructurar un entorn d'RA són: els **dispositius de visualització**, els **mètodes d'obtenció de dades de l'entorn** i les **interfícies amb l'usuari**.

2.1.1. Dispositius de visualització

Les aplicacions d'RA (com qualsevol altre aplicació) depenen fortament d'elements imposats pels seus requeriments funcionals i interfícies. Però les aplicacions d'RA es poden dissenyar per a dispositius molt diversos: són sistemes molt diferents que poden canviar considerablement els paradigmes de treball.

La classificació més genèrica que podem fer dels dispositius de visualització es basa en dos **models de presentació de la mescla de dades**: el model *see-through* o de projecció i el model vídeo. El **model *see-through* o de projecció** es basa en la superposició dels elements virtuals sobre la mateixa realitat. En canvi, el **model de vídeo** requereix d'un flux de captura que representi la realitat i els objectes virtuals s'hi insereixen per presentar la mescla a l'usuari en un display.

D'altra banda, també podem parlar d'una classificació de dispositius de visualització de RA segons la **forma de la interfície de visualització**; és a dir, mitjançant quin dispositiu l'usuari veu la mescla de realitat i virtualitat. Aquests dispositius són molt variats i poden anar des d'unes "ulleres" fins al més popular monitor d'ordinador personal, passant fins i tot per dispositius de visualització mòbils:

a) Les ulleres màgiques: HMD

El primer dels dispositius plantejats per treballar amb RA són els **Head Mounted Displays (HMD)**. Aquest dispositiu es basa en una mena d'ulleres que substitueixen els tradicionals vidres per un suport de display. Existeixen dos tipus diferents d'HMDs: els *See-through HMD* i els *Vídeo HMD*.

- **Vídeo HMD:** En aquest cas el display es compon per petites pantalles que reproduïxen les imatges capturades per una càmera (o preferiblement dues, per no perdre l'estereoscòpia de l'usuari). Afegir un element virtual implica un tractament d'aquesta imatge. El seu funcionament es pot veure representat a la figura 5a.
- **See-through HMD:** Aquestes ulleres es componen d'un vidre a través del qual l'usuari pot veure el món amb normalitat. Aquests vidres però, estan dotats d'unes característiques especials que permeten superposar-hi dades. El podem veure representat a la figura 5b.

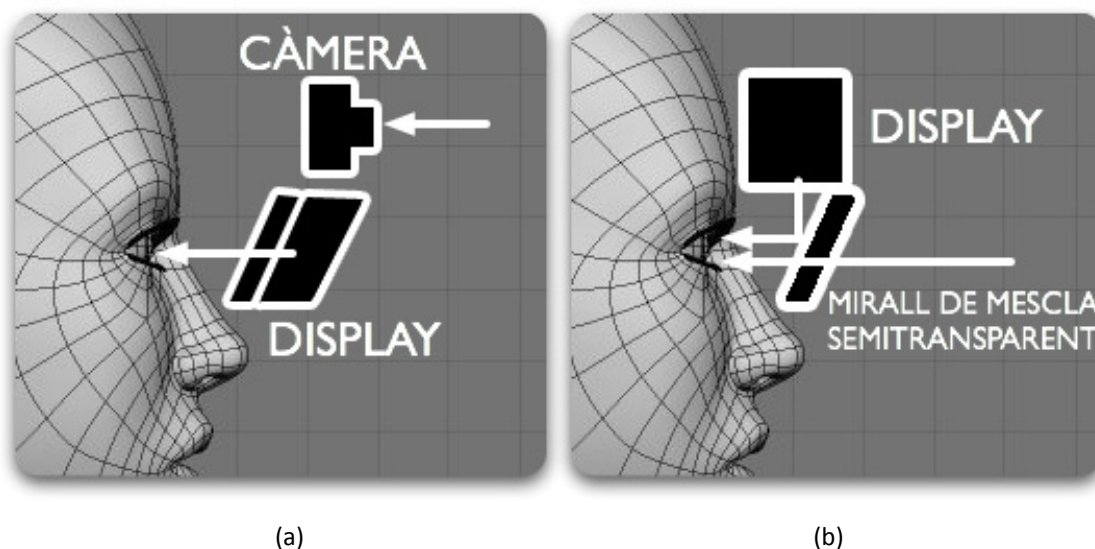


Figura 5: Esquema de funcionament de vídeo i see-through HMDs

Un dels avantatges més importants que ofereixen aquests displays és la possibilitat d'immersió de l'usuari en l'entorn augmentat. Posar-se unes ulleres permet veure un món augmentat de forma plenament "natural". Per altra banda, el principal inconvenient d'aquestes solucions rau en l'enfocament. En el cas del *see-through HMD*, l'ull de l'usuari és lliure d'enfocar diferents punts de l'espai i això en dificulta la representació dels elements virtuals sobre la lent, ja que aquests s'haurien d'adaptar als canvis d'enfocament. En el cas dels dispositius *vídeo HMD*, la restricció rau en el fet que les pantalles mostren imatges capturades per càmeres, imatges amb un punt fix d'enfocament. L'usuari veurà, en aquest cas, punts de l'espai desenfocats. També són significatius els problemes de calibració de les lents *see-through*, ja que han de tenir en compte on és i cap a on mira l'ull de l'usuari per poder alinear de forma correcta els objectes virtuals a la realitat. I, finalment, cal destacar que els dispositius *vídeo HMD* poden suposar un problema important per l'usuari: si s'apaguen o hi ha un problema en la captura de càmera, aquest quedaria cec.

Aquests mètodes de visualització solen ser fàcilment extensibles per oferir la possibilitat d'interaccionar amb el medi de forma natural; en la major part dels casos s'utilitzen interfícies dissenyades per interaccionar amb les mans, sons o moviments de l'usuari. No per això estan limitades a la utilització de dispositius externs que puguin ser utilitzats com a interfície (*joystics*, llapis òptics, teclats, etc.).

b) Les finestres cap al món augmentat: *Handhelds*

El segon dels dispositius de visualització són els *handhelds* (dispositius en mà). Aquests dispositius poden ser agafats per l'usuari com si fossin una finestra física i, a través seu, s'hi pot veure el món real amb els elements virtuals mesclats. Actualment es basen generalment en ser dispositius que treballen amb un flux de vídeo com els *video HMD*, però també n'existeixen basats en el model *see-through*. Aquests dispositius poden anar des d'uns binocles (idea molt pròxima als *HMD*), fins a telèfons mòbils i *PDA*s (figura 6) que presenten el món augmentat en els seus displays.

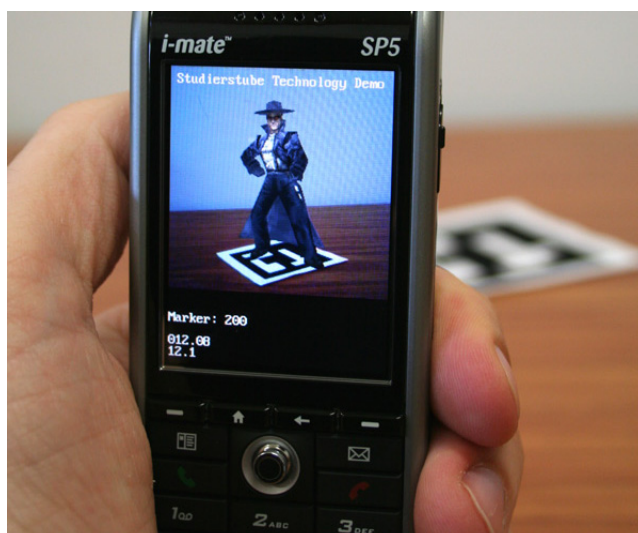


Figura 6: Realitat Augmentada a Handheld

La capacitat d'immersió que ofereixen aquests dispositius no és la mateixa, però poden ser una bona solució als diversos entrebancs que ara per ara suposen els *HMD*s. En molts casos també poden ser interessants per desenvolupar interfícies en les quals l'usuari pugui interaccionar directament amb el dispositiu.

Cal dir que aquestes solucions han proliferat molt en els últims anys en dispositius mòbils com telèfons i *PDA*s, degut a que aquests han augmentat considerablement la seva integració i disposen de *hardware* cada cop més potent (es preparen dispositius *embedded* que inclouen nuclis *ARM* de fins a 600MHz, hardware específic d'acceleració gràfica 2D/3D i elements de càlcul avançat pel tractament de vídeo i dades multimèdia) [16]. Aquest avenç tecnològic en el hardware dels dispositius mòbils ha fet possible la proliferació de solucions d'RA sobre *handhelds* i permetran millorar-ne la seva qualitat en els propers anys.

c) Un clàssic: el monitor

Basat en el model de vídeo com a representació de la realitat, el **monitor "clàssic"** (figura 7) és també una de les alternatives amb les quals es pot treballar amb RA. És molt comú perquè permet la implementació d'aplicacions d'RA sobre ordinadors personals i d'altres dispositius com televisions i, fins i tot, projeccions sobre pantalla.



Figura 7: Realitat Augmentada al monitor "clàssic"

Utilitzen una captura amb una càmera que pot ser fixa (amb la qual l'usuari interaccionarà modificant l'entorn) o bé mòbil (que pot comandar l'usuari per veure l'entorn des de diversos punts de vista). Els resultats s'obtenen en pantalla i permeten desenvolupar interfícies tant "naturals" com mitjançant el dispositiu.

Tot i no disposar d'un valor d'immersió com a les anteriors propostes, aquesta solució és molt potent per la seva simplicitat, versatilitat i per la capacitat de còmput dels ordinadors personals; a la vegada que demostra que no és necessària la immersió per crear Realitat Augmentada.

d) Projecció: elements virtuals literalment al món real

Per últim, no volem ometre una última manera de veure la realitat augmentada: "la **projecció** d'objectes virtuals". Tot i no disposar fins a dia d'avui de resultats molt satisfactoris, hi ha investigacions que proposen la utilització de la projecció sobre l'entorn dels objectes virtuals. En molts casos això implica la limitació de que només pugui veure-ho un usuari, o bé treballar amb materials especials i/o displays que ofereixin reflexions diferents per a cada angle de visió. En un futur, podria derivar també en solucions que implementin la visualització d'hologrames.

2.1.2. Obtenció i tractament de dades de l'entorn

La segona de les característiques a tenir en compte és **seguiment i l'alineació** (*tracking and registering*). Per tal de poder aplicar les imatges virtuals a la realitat com si en formessin part, abans que res, necessitem aconseguir dades del nostre posicionament respecte al món i traduir-ho per tal d'incloure els elements virtuals de forma satisfactòria. Podem obtenir dades de **seguiment** per mitjà de sensors, visió per computador o sistemes mixtos.

a) Sensors

Podem obtenir aquestes dades amb un llarg etcètera de sensors: via un sistema de posicionament global (GPS o Galileu), sensors inercials que ens poden informar sobre com es mou un dispositiu, brúixoles electròniques que ens permeten conèixer l'orientació en la que es troba un dispositiu. Aquests dispositius poden obtenir un posicionament molt complet, però per fer-ho, requereixen treballar combinats i amb mecanismes que en suavitzin la seva "sorollosa" recepció.

b) Visió per computador

Gràcies a la interpretació d'imatges podem modelar com és el món que veiem i utilitzar aquesta representació per afegir-hi els elements virtuals. De forma generalitzada, aquest mètode es pot dividir en dues tècniques:

- **Seguiment de punts de referència explícits.** Els punts de referència explícits es solen modelar pel seguiment de patrons a l'escena (*fiducial markers*), interpretant-ne la seva posició, i utilitzant-los com a orígens de referència per l'espai virtual. Exemple a la figura 8a.
- Per l'altre banda, podem agrupar els **seguits per punts de referència no explícits.** En aquest cas, l'algoritme de visió per computador no té elements específics a aïllar i utilitzar com a marcador. Fa servir característiques de l'entorn per estudiar el posicionament i situar-hi els elements virtuals. Figura 8b.



Figura 8: Seguiment explícit d'un marcador i seguiment per punts característics de l'escena

c) Sistemes mixtos

Els sistemes mixtos inclouen conjuntament diversos sensors i/o visió per computador. En moltes ocasions, les investigacions fetes fins a dia d'avui mostren els avantatges d'utilitzar sistemes mixtos per aconseguir que les dades siguin més estables i dissenyar així sistemes més robustos.

Cal notar que errors de càlcul o imprecisions de la captura de dades fan que els algoritmes de seguiment no siguin capaços de respondre sempre en temps real i, per això, s'acompanyen d'algoritmes que permeten fer **prediccions** i així complementar el flux de dades.

Finalment, el seguiment es complementa amb l'**alineació**, que en traduirà les dades resultants per tal d'obtenir el posicionament del punt de vista de l'usuari respecte al món i construir l'escena augmentada. En donarem més detalls en el proper capítol de la memòria.

2.1.3. Interfície amb l'usuari

L'última de les característiques que s'han de valorar és la interfície que es pot oferir. Aquest és un aspecte que varia molt entre aplicacions, però hi ha certs elements destacables que es poden distingir a totes els entorns d'RA.

Hi ha aplicacions que es podrien considerar **sense interfície** d'interacció directa: l'usuari és només un espectador, però un espectador amb la possibilitat d'interactuar definint, per exemple, cap a on mira amb un *HMD*.

En certes aplicacions, com comentàvem a l'apartat de visualització, s'utilitzen **dispositius externs** per complementar la interacció; n'és un exemple un entorn que treballa amb un *HMD* i disposa d'un *joystick*, teclat o similars per enviar comandes a l'entorn. Però aquestes interfícies són poc intuïtives i s'intenten millorar aconseguint una interacció més natural. El primer exemple el podem trobar en l'article *Developing a Generic Augmented Reality Interface* [17] que proposa un **mètode tangible**: utilitzar marcadors (fiducials) plans que puguin ser agafats i moguts per l'usuari. Cadascun d'aquests és la base d'un element virtual. Aquests elements fins i tot poden interactuar entre ells per proximitat, per gestos apresos o simplement per la seva aparició en escena. En aquesta mateixa línia existeixen altres aproximacions que, enlloc d'utilitzar plans, exploren la interacció amb cubs o altres poliedres amb dibuixos marcadors o fins i tot sense ells. Són sens dubte les interfícies més explorades en la Realitat Augmentada; les utilitzen les llibreries més conegudes com *ARtag* [15] i *ArtoolKit* [14], però també han estat triades per aplicacions comercials com la ja esmentada *Eye of Judgement* de Sony [13]. Una altra vessant intenta explotar al màxim la **interfície natural**: estudiar les extremitats de l'usuari ja sigui mitjançant visió per computador o guants amb sensors (figura 9). Així es fan servir els moviments per traduir-los amb interacció amb els elements virtuals. Aquestes tècniques es solen complementar amb perifèrics *haptics* que puguin generar sensació de contacte.



Figura 9: Guants sensors de gestos amb interfície hàptica

2.2. On s'emmarca aquest projecte?

Un cop estudiades totes les alternatives de disseny, per aquesta proposta decidim fixar els nostres objectius dins de les propostes vistes: Pel que fa a la **visualització**, hem escollit treballar amb el model de **monitor "clàssic"** per motius de simplicitat. El monitor ens oferirà versatilitat suficient per treballar amb qualsevol dels algoritmes de visió per computador existents a més d'evitar de requerir de hardware addicional amb el cost que pot suposar. A més a més, aquest model és suficient per el desenvolupament d'un entorn de desenvolupament i podria ser exportat a un altre dispositiu com per exemple *HMD* amb un cost d'implementació relativament baix ja que és també un dels models basat en captura de vídeo. Ens decantem pel model d'**interfície tangible** amb marcadors donada la seva gran versatilitat a l'hora de treballar amb tot tipus d'aplicacions d'RA, perquè és una bona opció de disseny per la interacció amb l'usuari, i permet fer el **seguiment i alineació** mitjançant **visió per computador**.

Capítol 3

Anàlisi, disseny i implementació

En el capítol anterior hem explorat els espais de treball en els que pot estar emmarcat el disseny d'un entorn de Realitat Augmentada i hem pogut decidir a quins d'ells ens adaptarem per realitzar el nostre projecte. En aquest capítol avançarem cap a la **implementació de l'entorn** bàsic sobre el qual treballarem més endavant en una **solució** específica d'RA. Abans d'endinsar-nos en aquest desenvolupament però, farem un **estudi de viabilitat** del nostre plantejament i en realitzarem una **anàlisi** que ens permeti realitzar el **disseny de l'entorn i l'aplicació**.

3.1 Estudi de viabilitat

En resum, en aquest projecte implementarem un entorn de desenvolupament i proves de Realitat Augmentada. Aquest entorn té requeriments que hem d'analitzar amb l'objectiu de determinar la viabilitat de la nostra proposta abans d'aprofundir-hi en més detall.

Com hem comentat, basarem el projecte en el llenguatge *C++* per motius de rendiment. Això, en certa manera implica una complexitat elevada per ser un entorn de proves, però ens recolzarem en diverses llibreries per facilitar aquest desenvolupament. Ens cal treballar amb gràfics 3D, realitzar una captura de vídeo i també treballarem amb tractament d'imatges i visió per computador:

- Pel desenvolupament gràfic tridimensional ens recolzarem la llibreria *OpenGL* [18],
- pels tractaments d'imatge i tècniques de visió per computador s'utilitzarà la llibreria *OpenCV* [19],
- i finalment, per la captura d'imatges ens basarem amb la llibreria multimèdia *DirectShow* [20].

Les dues primeres llibreries (*OpenGL* i *OpenCV*) són llibreries de software lliure i gratuït. Accedir a qualsevol de les funcionalitats que implementen no suposarà doncs, cap problema. De la primera en tenim coneixements pel seu ús al llarg de la carrera en les assignatures de "Gràfics per Computador". En el cas d'*OpenCV*, a l'inici d'aquest projecte disposem de rudiments sobre la seva funcionalitat bàsica que ens ajudaran a introduir-nos-hi amb més facilitat. En principi, adaptar-se a qualsevol dels àmbits desconeguts d'aquestes llibreries no hauria de ser un problema perquè disposem de documentacions oficials [21][22] accessibles per l'usuari, així com de grups de discussió en línia com el de *Yahoo!* per *OpenCV* [23] on els seus usuaris col·laboren per solucionar problemes.

Pel que fa a *DirectShow* (que serà la llibreria encarregada d'ajudar-nos en procediment de captura d'imatges), també disposem de coneixements previs obtinguts durant l'enginyeria a l'assignatura de "Sistemes Experts".

Aquests coneixements i la seva documentació completa per part de *Microsoft* a *MSDN* [24] són el motiu que ens ha fet triar-la per iniciar el desenvolupament en el menor temps possible.

En darrer lloc, aquest projecte està enfocat a treballar amb un ordinador personal, amb potència suficient per executar els algoritmes que el composin i que disposi de hardware específic d'acceleració gràfica (totalment imprescindible per treballar amb *OpenGL* de forma satisfactòria).

L'equip disponible per treballar en el desenvolupament i amb el qual es realitzaran totes les proves comentades en aquesta memòria és un ordinador portàtil amb processador *Intel Core 2 Duo 2,2GHz* [25] amb 1Gb de RAM i hardware d'acceleració *NVidia Geforce 8400Mobile GS* [26] amb memòria dedicada de 128Mb; una potència en principi suficient per treballar-hi. En cas de no ser suficient es pot recórrer a d'altres equips més potents. Finalment, el *hardware* de captura de vídeo disponible és una webcam *Logitech QuickCam Pro 4000* [27] amb resolució 320x240 píxels amb capacitat d'exportar 30 captures per segon; suficient per construir un entorn en temps real. També es podran fer proves amb la càmera integrada a l'equip portàtil que disposa d'unes característiques similars.

3.2 Anàlisi de requeriments

Fins ara tant sols hem vist quines son les principals característiques que defineixen les fronteres de cadascun dels entorns possibles que poden definir un entorn de Realitat Augmentada. Ara cal lligar aquests i d'altres elements més específics que ens permetran conèixer el flux de treball del nostre entorn, per així estudiar-ne més a fons el seu funcionament i organització:

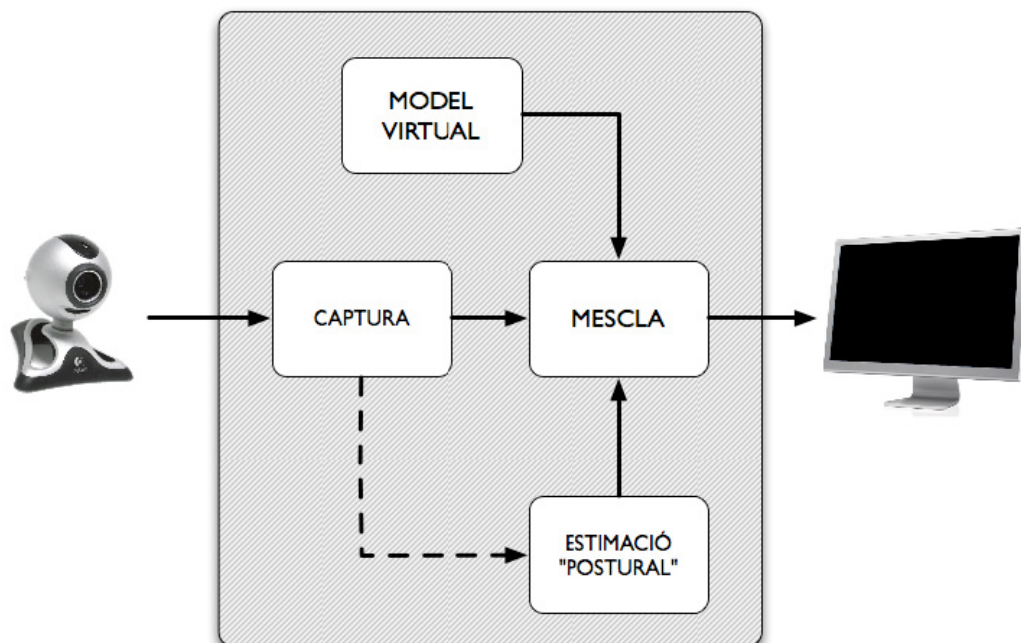


Figura 10: Esquema del flux de treball d'un entorn genèric de Realitat Augmentada basat en Visió per Computador

Observant el procés genèric de construcció d'una escena de Realitat Augmentada (escena augmentada), podem descriure de forma esquemàtica el flux de treball com indica la figura 10.

El procés comença en la **captura d'imatges** per part d'una càmera connectada a l'equip sobre el qual s'executarà l'entorn. En un entorn orientat a ser basat en un model de vídeo, les imatges provinents d'aquest dispositiu s'obtenen per tal de formar "la part real" en la composició final. Cal aclarir que a aquest entorn se'l podria fer treballar amb qualsevol altre font contínua de dades de vídeo susceptibles de formar part d'un procés d'augmentació. Per exemple: *streaming* televisiu, enregistraments emmagatzemats a l'equip o bé en xarxa, videoconferències en temps real, etc.

D'aquesta captura, amb un mètode d'obtenció de dades mitjançant visió per computador, les imatges successives que la componen podran ser **tractades** per un seguit d'algoritmes per obtenir una estimació de la posició de la càmera respecte al món (en el cas del nostre projecte: de la càmera respecte als marcadors). Aquesta estimació s'enviarà a les funcionalitats de mescla. En cas de tractar amb un model d'obtenció de dades a partir de sensors, aquest no hauria d'accedir al flux de imatges i enviaria les seves estimacions a les funcionalitats de mescla.

El **procés de mescla** serà l'etapa en la qual es composaran els **models virtuals** amb la captura de vídeo original per construir l'escena augmentada que es mostrarà a l'usuari en el **pantalla**.

Per implementar aquestes operacions hem estructurat el desenvolupament en dues parts, fent una separació entre el mètode d'estimació de posicionament i la resta de l'entorn. El fet d'independitzar els dos processos (figura 11) es deu principalment a l'objectiu d'aquest projecte: crear un entorn de proves. Anteriorment hem esmentat que en l'apartat del "posicionament" hi ha molta varietat d'algoritmes, però la majoria de les implementacions existents a dia d'avui son molt específiques i s'adapten a cadascuna de les aplicacions finals d'usuari. Per això és important independitzar l'obtenció de dades de posicionament dels models virtuals de la resta de l'entorn, ja que se li vol donar el millor disseny per poder implementar-hi proves amb facilitat. Així doncs, primer desenvoluparem l'**entorn** dedicat a la captura, mescla de realitat i virtualitat, i per altre banda articularem una **solució d'RA**¹ específica basada en visió per computador per obtenir l'estimació de posició per situar els objectes virtuals.

En els últims anys, hi ha hagut alguns intents de definir patrons de disseny software pensats per construir entorns de Realitat Augmentada. N'és un bon exemple l'article *Towards a System of Patterns for Augmented Reality Systems* [28] de la Universitat de Múnchen, Alemanya. En aquest article es presenta d'una forma estructurada una proposta de disseny genèric en la qual ens hem inspirat per a construir aquest entorn. La separació conceptual que fem entre "entorn" i "solució" no es té en compte en aquesta proposta ja que és una restricció imposada al crear un entorn de proves, però sí es tindran en compte altres detalls per realitzar els dissenys d'implementació.

¹ Per **solució d'RA** ens referim a la implementació d'un algoritme d'estimació prototipus que servirà per validar el nostre entorn i a la vegada permetrà desenvolupar aplicacions prototipus basades amb l'entorn com si es tractés d'una llibreria de desenvolupament de Realitat Augmentada.

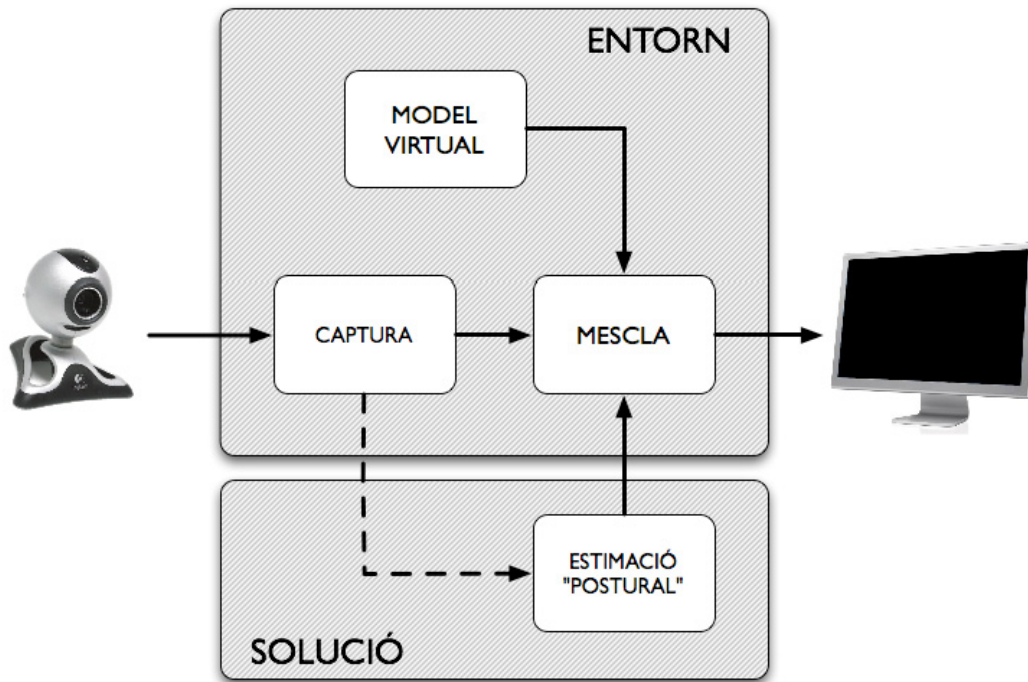


Figura 11: Flux adaptat per l'entorn de proves

3.3 Disseny de l'entorn

Hem volgut agrupar en un mòdul independent totes aquelles tasques que són menys susceptibles de necessitar canvis, ja que no formen part de l'objectiu del desenvolupament proves de prototipus d'algoritmes i aplicacions basats en Realitat Augmentada. Aquest primer mòdul s'encarregarà doncs, de la gestió del flux de vídeo provinent de la càmera i de la mescla entre elements virtuals. Tot i que l'entorn no en forma part del concepte presentat, l'inclourem en una interfície gràfica d'usuari per realitzar-ne la presentació en pantalla i les proves.

L'entorn s'ocupa de la gestió de captura i per això també serà l'encarregat de facilitar l'accés directe al flux de vídeo a mòduls externs que es puguin basar en visió per computador (com la solució d'RA que implementarem per validar aquest entorn). També haurà d'habilitar l'accés a la informació relativa a posicionament calculada en aquests mòduls externs per completar el seu processament amb la mescla. Tot i la separació que fem entre "entorn" i "solució", cal dir que l'entorn també s'estructura en diversos mòduls per facilitar-ne la seva comprensió i les possibles modificacions que puguin sorgir en diversos tipus d'implementació. Es realitza la captura d'imatges en un mòdul independent per aïllar l'entorn de possibles canvis que poguessin introduir-se en el seu funcionament (com podria ser l'adaptació de l'entorn a d'altres sistemes operatius), i per d'altra banda s'agrupa la funcionalitat de mescla d'imatges i elements virtuals. L'arquitectura presentada es pot veure a la figura 12. A continuació en descriurem els detalls referents a la **captura d'imatges**, a la **mescla d'objectes virtuals i realitat**, a més a més del mecanisme d'**accés al flux de vídeo**.

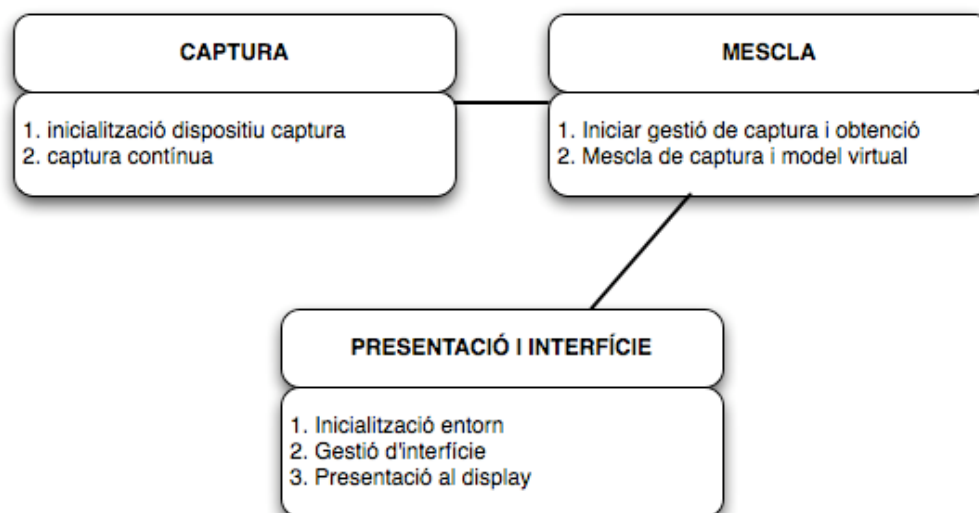


Figura 12: Esquema del disseny de l'entorn i l'aplicació d'interfície

3.3.1 Captura d'imatges

En la captura de vídeo, el punt de partida del projecte és utilitzar la captura d'un dispositiu *Logitech QuickCam Pro 4000* [27]. Aquest dispositiu és una **webcam** que permet capturar imatges amb una resolució de 320x240 a 30fps, velocitat i mida suficients per treballar en l'entorn que desenvolupem. La característica que l'ha fet més desitjable davant d'altres dispositius disponibles és que la seva captura és la més nítida. Aquesta captura, a nivell de software, es farà mitjançant les utilitats que ofereixen les llibreries *DirectShow* [20] de *Microsoft*. Aquest és un procés relativament senzill sobre l'entorn de desenvolupament *Microsoft Visual C++* [29], que pretenem que sigui la plataforma inicial de desenvolupament. A més a més, ja disposem d'experiència amb aquestes eines i aquest factor ha d'accelerar el procés de posada en marxa. Existeixen altres alternatives que seran avaluades en temps d'implementació de l'entorn i permetrien una senzilla migració a altres plataformes com *GNU Linux* o d'altres.

A priori es pot pensar que treballar amb un llenguatge de programació com *C++* pot implicar certa complexitat a l'hora de generar un prototipus com aquest. De fet, en molts projectes es sol recórrer a d'altres eines com *Matlab* que permeten desenvolupar els prototipus en menys temps. Aquests però, generalment no permeten obtenir entorns d'alt rendiment. L'entorn que proposem per a aquest projecte tindrà uns **alts requeriments de còmput** per tal d'oferir un bon rendiment: cal recordar que estem parlant d'una construcció en temps real que tracta i mescla dades de vídeo i gràfics 3D. Per adaptar-nos a aquesta necessitat requerim des de bon començament l'ús d'un llenguatge de més baix nivell com *C++*, amb el qual podem aspirar a un millor rendiment.

3.3.2 Mescla d'objectes virtuals i realitat

Per poder superposar els objectes virtuals d'una forma tridimensional sobre l'escena real (recordem que aquest és un requeriment imprescindible a l'hora de generar Realitat Augmentada), primer ens cal situar-los en una postura coherent amb la imatge capturada. Aquesta postura l'estimarem amb un mòdul extern (en el nostre projecte, a la solució d'RA que desenvoluparem a continuació de l'entorn) i s'hi hauran de fer les transformacions

adients per tal de col·locar els elements virtuals a aquesta estimació. Aquesta, serà la relació que hi ha entre el pla de captura i el pla d'origen sobre el que s'inscriu l'objecte virtual, i es pot considerar recorrent a dos models: un model centrat en la càmera i un model centrat en l'objecte, representats a la figura 13.

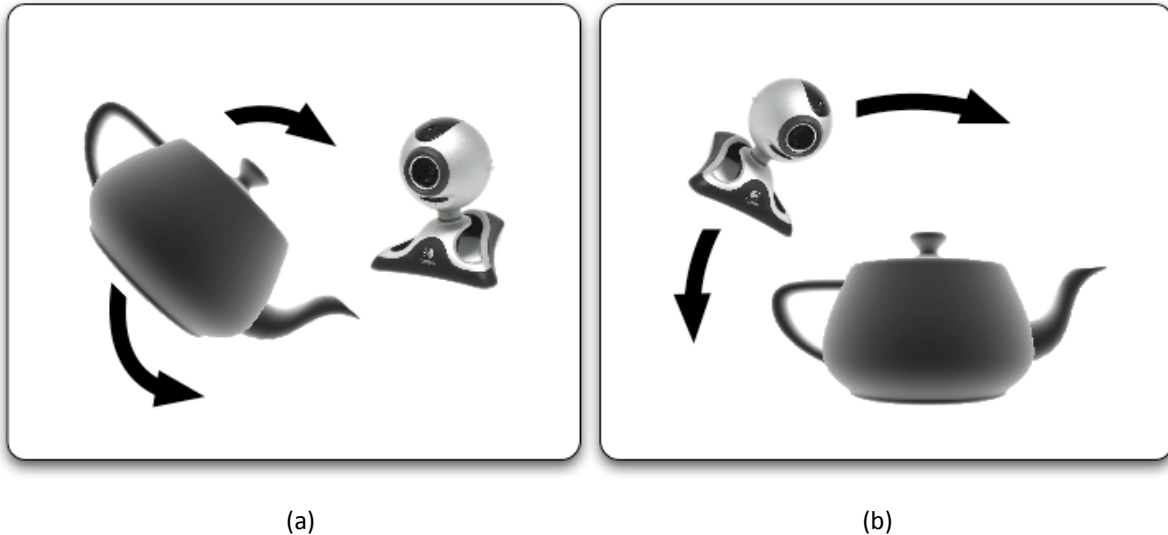


Figura 13: Representació esquemàtica dels models "centrat en la càmera" i "centrat en l'objecte"

En el model **centrat en la càmera**, l'objecte es mou per adequar-lo a la postura final. La càmera és estàtica i no serà traslladada ni orientada de cap altra manera, serà l'objecte qui canviarà de posició. Per altre banda, el model **centrat en l'objecte** planteja el problema de forma inversa. En aquest cas, l'objecte és estàtic i s'adequarà la seva posició per la rotació de la càmera al seu voltant.

En *OpenGL*, llibreria triada pel desenvolupament tridimensional, aquests dos models de treball no haurien de generar diferències de còmput, de manera que es pot triar en temps de desenvolupament quin és el model més intuïtiu i més senzill per treballar amb les dades disponibles.

Després de situar el model virtual acord amb la seva posició final al món real i disposant de la imatge capturada, procedirem a **composar l'escena augmentada**. Aquesta operació la realitzarem també amb la llibreria *OpenGL*. Cal destacar com a part del disseny que existeixen dos mètodes per enfrontar aquest problema:

- El primer es basa en utilitzar la captura com a textura aplicada a un rectangle estàtic que observa la càmera.
- L'altre mètode utilitza aquesta captura per copiar-la "directament" cap al *buffer d'OpenGL*.

El rendiment d'aquestes operacions pot ser molt diferent i és un problema que s'ha de solucionar avaluant ambdues alternatives. Hi entrarem amb més detall a l'apartat de desenvolupament, però és important veure que és especialment necessari obtenir el millor resultat de rendiment en aquesta operació ja que podria arribar a ser un potencial coll d'ampolla de l'entorn. Si ens limita la velocitat de representació podríem arribar a no comptar amb una augmentació en temps real que implicaria un replantejament del projecte.

3.3.3 Accés al flux de vídeo

L'entorn estarà preparat per utilitzar el flux de vídeo a més de compartir-lo amb els mòduls externs d'estimació de posició que pretenguin utilitzar-lo per realitzar operacions d'estimació mitjançant visió per computador o mètodes mixtos. Aquesta compartició es farà de forma totalment transparent a nivell de "format" de les dades, compartint la informació en model *bitmap*. Per tant, la traducció de format al que la solució externa necessiti es farà fora de l'entorn per permetre així la màxima llibertat dels mòduls extens. Com veurem, la nostra solució d'RA utilitza la llibreria *OpenCV*, però altres solucions podrien recórrer a d'altres llibreries diferents (*VXL*, *VIGRA*, *NokiaCV*, *LTI-Lib*, etc.) i fins i tot mètodes especialment desenvolupats en *C++*.

3.4 Implementació de l'entorn

En aquest apartat de la memòria explicarem els detalls que apareixen en el procés de desenvolupament de l'entorn: quins han estat els passos seguits per arribar a la solució final i perquè han estat triats en cada moment. Els desglossarem en desenvolupament de la **interfície i presentació**, desenvolupament de la **captura d'imatges i gestió del flux de vídeo i mescla d'objectes virtuals i realitat**.

3.4.1 Interfície i presentació

Per tal de crear l'espai de representació en pantalla, *OpenGL* disposa d'una part de la llibreria (*GLUT: GL Utility Toolkit*) [30] preparada de forma que integra en sí mateixa la presentació en finestres al sistema operatiu. Aquestes primitives doncs, permeten construir un entorn molt senzill sense la necessitat d'utilitzar altres sistemes externs per integrar aquesta visualització en finestres. Per tot això, iniciem l'etapa de desenvolupament amb aquesta utilitat de la llibreria.

Cal afegir que *GLUT* dificulta el treball amb classes perquè no pertany a un disseny orientat a objectes, tot i que s'hi poden integrar. A més a més, si volíem afegir algunes eines d'interfície gràfica d'usuari per permetre accedir a la sintonització de paràmetres interns del nostre programa, *GLUT* disposa d'un repertori molt limitat que es basa únicament en menús contextuals. Això va fer que, en relativament poc temps, prescindíssim de les utilitats *GLUT*. D'aquesta manera podríem integrar aquests controls. El desenvolupament inicial s'estava fent sobre *Windows XP*, el qual va permetre l'ús de la llibreria de creació de *GUIs* de *Microsoft*, *MFC*. Es tracta d'un entorn més complex, però va servir per poder treballar amb molta més comoditat a l'hora de programar correctament la part de visió per computador que requeria de constants sintonitzacions en els seus paràmetres interns. Aquest canvi també va servir per estructurar el programa en un entorn purament orientat a objecte.

3.4.2 Captura d'imatges

El primer pas a desenvolupar pel projecte és la captura de vídeo. Ens decantem per utilitzar les llibreries multimèdia *DirectShow* perquè iniciem el projecte sobre *Windows* i tenim coneixements sobre les mateixes.

DirectShow és el sistema que utilitza *Microsoft* en els seus sistemes operatius *Windows* per gestionar els fluxos de dades multimèdia. A grans trets, *DirectShow* es basa en una interconnexió (graf) de mòduls que generen, processen i presenten aquestes dades. Aquests mòduls formen part de la tecnologia *COM* (*Component Object Model-based*) [31] que implementa un model modular abstracte i de caixa negra que permet utilitzar els seus mòduls fins i tot distribuïts per xarxa.

Una de les solucions emprades per treballar amb *DirectShow* i *OpenCV* és construir tota l'operativa *OpenCV* internament a un mòdul *COM* que formi part d'un graf. Això però, té l'inconvenient d'estar creant un programa basat en una entitat abstracta, el qual comporta una complicació innecessària en el nostre projecte i limitacions per implementar entorns complexos. Per la nostra escomesa volem simplificar aquest problema. Hem construït un graf que obté les imatges provinents de la càmera i les posa a disposició en una interfície de sortida del mateix.

Per aconseguir aquesta implementació construïm un graf *DirectShow* com el que mostra la figura 14.

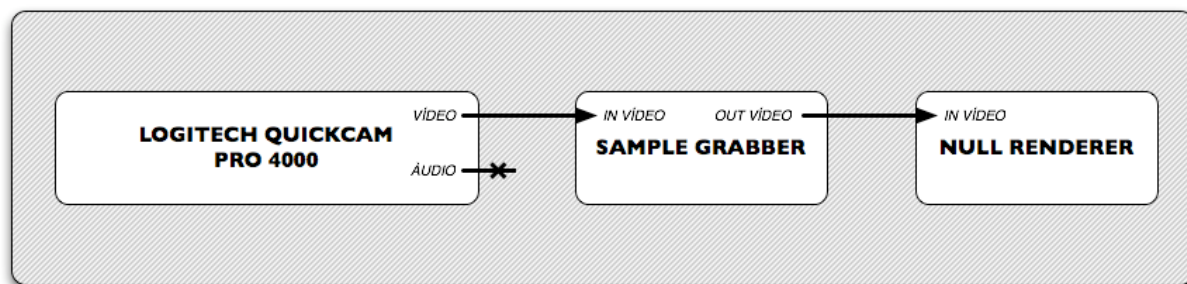


Figura 14: Graf implementat amb *DirectShow* per a la captura d'imatges des de la webcam

Els **controladors de càmera** a *Windows* són mòduls *COM* per a *DirectShow*; podem veure que el controlador de la nostra càmera (*Logitech Quickam Pro 4000*) és el primer mòdul d'aquest graf. Serà l'encarregat de comunicar-se amb la càmera: encendre-la, configurar-ne els paràmetres principals i iniciar el flux de captura.

El següent mòdul al qual va a parar el flux de captura és l'anomenat **SampleGrabber** [32], un mòdul inclòs en la llibreria que permet fer captures puntuals o contínues del flux de captura. El configurarem en mode de captura contínua i activarem una **interfície de sortida** amb la qual ens comunicarem des del nostre programa per recuperar les imatges capturades.

El mòdul *SampleGrabber* de la llibreria està pensat per ser introduït en un graf que pugui disposar d'aquestes dades a posteriori, així que en redirigeix l'entrada a la sortida. Aquesta sortida però, no pot quedar lliure, de manera que enviarem les dades al mòdul **NullRenderer**. Aquest no tracta les dades però tanca el graf de *DirectShow*.

3.4.3 Gestió del flux de vídeo i mescla d'objectes virtuals i realitat

La captura del vídeo és només el començament del treball de l'entorn. A continuació veurem com es fa la **gestió de vídeo** en l'entorn, però no ho podem fer sense descriure abans com es fa **pas del flux de captura a l'espai tridimensional**. Un cop realitzat aquest pas, detallarem com s'implementa la **superposició d'elements virtuals** i la **construcció de l'escena augmentada**.

a) Pas del flux de captura a l'espai tridimensional

El procediment d'unió dels elements virtuals i de la captura real és fonamental pel correcte funcionament de l'entorn. En cas que no haguéssim aconseguit fer que fos operatiu en *OpenGL*, s'hauria d'haver buscat alguna alternativa viable. Per això era necessari implementar-ho i poder determinar que el projecte era factible tal com estava plantejat en un inici.

Abans de res, volem portar la captura del vídeo realitzada en el pas anterior a l'espai 3D d'*OpenGL*. Com hem comentat en l'apartat d'anàlisi de requeriments, existeixen dues maneres de solucionar el problema i poder tenir rendiments molt diferents. Per això vam posar a prova ambdues alternatives:

- La primera prova es basa un polígon rectangular visible des del punt de visualització triat a l'espai 3D. L'objectiu és assignar dinàmicament les imatges provinents del dispositiu de captura com una textura del polígon. Per mantenir la fluïdesa de visualització, aquest *mapping* ha d'estar format per un mínim de 24 imatges per segon. Però això no va ser possible: el seu rendiment, en el millor dels casos, no superava les 12 imatges/segon en la nostra implementació.
- La segona d'ambdues proves és un procediment de bolcat d'imatge sobre la memòria de treball d'*OpenGL*. En aquest cas, s'inicialitza l'espai 3D amb una mida de visualització igual que la resolució de càmera i s'hi copien punt a punt les imatges capturades al *buffer* de treball d'*OpenGL*. En la nostra implementació, aquest procediment ofereix un rendiment molt alt: podem arribar a generar fins a un refresc màxim de 68 imatges per segon. La càmera en dóna tant sols 30, de manera que la resta d'imatges són duplicades i no milloren la fluïdesa. Cal tenir en compte però, que és especialment interessant disposar d'un rendiment molt superior al mínim, ja que en aquest rati encara manca descomptar el cost temporal de la resta d'operacions a realitzar amb cada frame.

Cal aclarir que aquests rendiments no depenen especialment de la implementació software. Si bé poden variar mínimament segons les implementacions de les llibreries gràfiques utilitzades, el que més pot afectar és el hardware d'acceleració gràfica utilitzat, ja que pot implementar aquestes operacions d'una forma més o menys òptima segons s'hagi considerat en el disseny dels processadors gràfics. Els rendiments mostrats són sobre l'equip comentat al estudi de viabilitat i la seva acceleradora gràfica.

b) Gestió del flux de vídeo

Com hem introduït al disseny, el nostre entorn comparteix la captura de vídeo amb els mòduls externs d'estimació de posició el flux de dades per construir entorns basats en visió per computador paral·lelament amb el seu ús per a la representació augmentada. Descriurem a continuació el funcionament final de la implementació d'aquesta gestió. Per fer-ho prendrem com exemple algun detall la solució d'RA que acompanyarà aquest entorn ja que tot i que no de forma directa, en forma part.

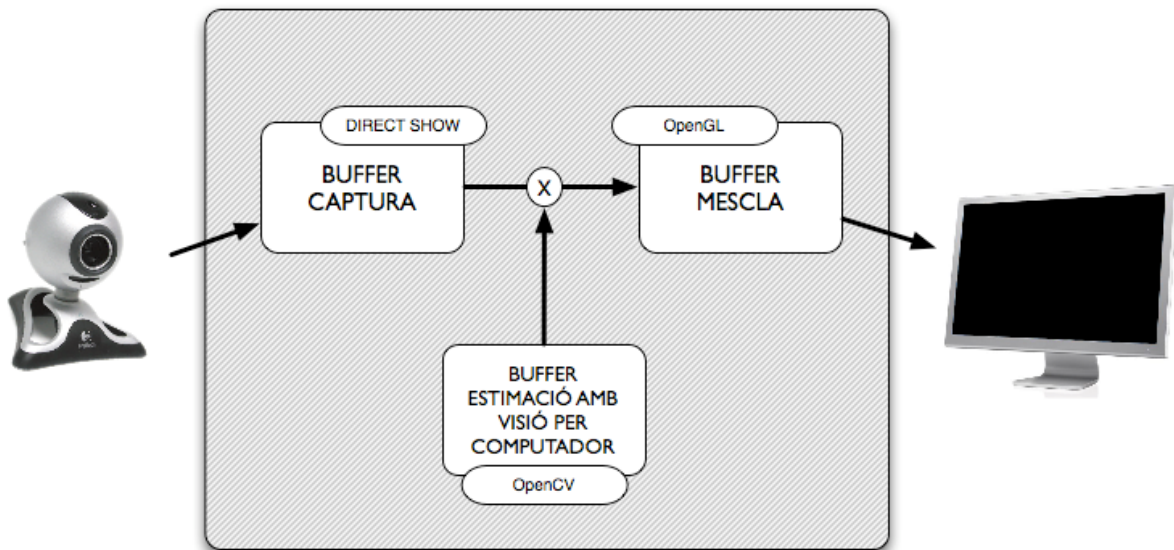


Figura 15: Gestió del flux de vídeo

A la figura 15 es descriu aquest procediment de forma esquemàtica. Volem portar el flux de vídeo al *buffer* d'*OpenGL* i que el mòdul extern pugui accedir a aquest flux de vídeo (en la nostra solució significarà fer un pas de la captura cap al *buffer* de treball d'*OpenCV*). Per simplificar les tasques de desenvolupament, hem vist que és molt interessant la possibilitat de fer modificacions sobre les imatges des d'*OpenCV* i retornar-les per ser utilitzades com "la part real" en la mescla. Això permet fer *debugging* gràfic, observant en l'entorn *OpenGL* les operacions d'*OpenCV*. Es tracta d'una unió conceptual entre els buffers de captura a *DirectShow*, de presentació a *OpenGL* i de còmput per a *OpenCV*.

c) Superposició d'elements virtuals al flux de vídeo

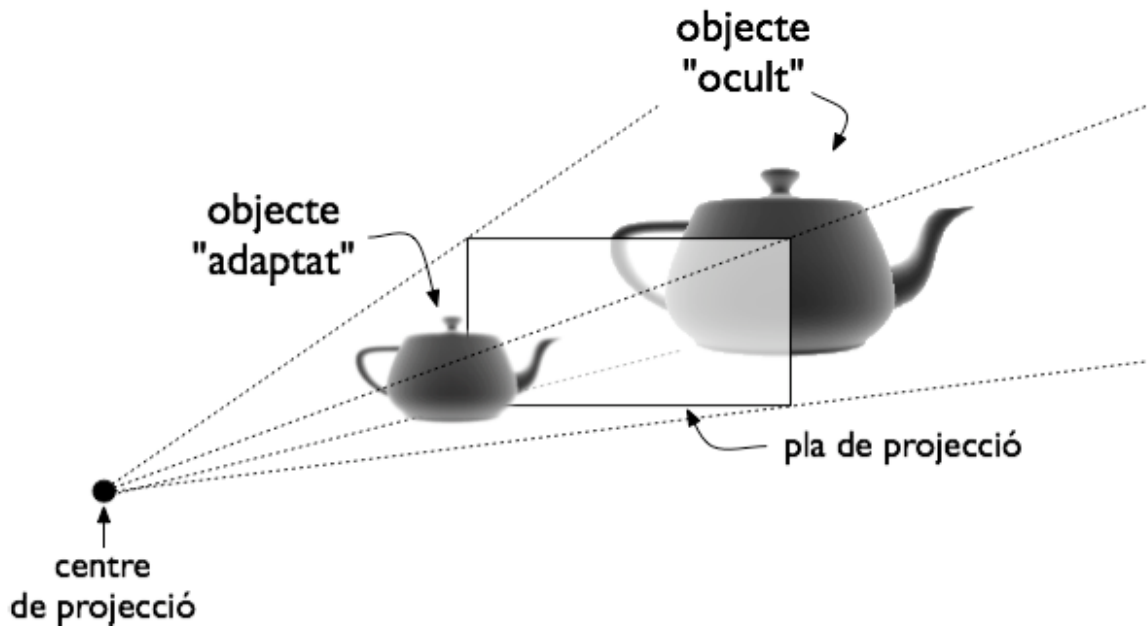
Un cop aconseguim disposar del flux de vídeo integrat a l'espai *OpenGL*, queda pendent la integració de les parts virtuals per completar aquesta etapa. En aquesta operació hem de fer visibles els objectes virtuals superposats al flux de vídeo. Això es pot resoldre de dues maneres:

- **adaptant el model virtual:**

La primera opció és adaptar el model virtual a una mida determinada per poder situar-lo entre el pla de projecció i el centre de projecció (figura 16: objecte "adaptat"). Aquesta operació implica fer còmputos afegits per adequar el model. Ho podem evitar amb el següent mètode.

- **controlant els càlculs de profunditat i ocultació** (figura 16: objecte "ocult"):

En aquest cas, es pot jugar amb una propietat dels espais tridimensionals: la profunditat. Per poder fer les representacions, la llibreria gràfica calcula la profunditat dels elements i en determina així la seva visibilitat. Aleshores, només cal desactivar el test de profunditat abans de dibuixar l'escena virtual: així no es considerarà cap ocultació i quedarà en primer pla per davant del vídeo. L'escena però, al ser dibuixada, requereix d'ocultació per garantir la correcta seva representació. Això s'aconsegueix dibuixant aquesta escena en un context a part que podem crear en *OpenGL* en el que sí estigui activat el test de profunditat.



Figures 16: Modes de superposició dels models virtuals

d) Construcció de l'escena augmentada

Disposant del mètode per superposar els elements virtuals en el nostre espai tridimensional, i rebent la informació relativa al posicionament dels elements virtuals a l'escena que es calcularà en la solució d'RA externa a l'entorn, arribem a l'etapa final del processament de l'entorn: la construcció de les escenes augmentades.

El mòdul extern ens haurà de facilitar les dades que ens relacionen el punt de vista i el patró identificat (paràmetres extrínsecs; rotació i translació). També haurà de donar-nos les dades sobre la captura (paràmetres extrínsecs; distorsió del dispositiu de captura sobre la realitat) que haurem d'aplicar per millorar la precisió de representació dels elements virtuals d'acord amb la realitat capturada.

Al treballar amb *OpenGL* com a llibreria gràfica, aplicar aquestes operacions requerirà tant sols una matriu 4x4 de visualització tridimensional (que aplicarem per tal de situar el punt de vista sobre el món virtual) i una matriu 4x4 de projecció (que indiqui els paràmetres amb els quals ha de treballar la "càmera virtual"). En donarem més detalls en la solució d'estimació "postural" presentada a continuació.

3.5 Disseny de la solució d'RA

El nostre entorn per si sol només implementa l'operativa bàsica que gestiona vídeo i 3D. L'objectiu d'aquesta separació és que es puguin posar a prova algoritmes independents de l'entorn que permetin completar l'escena augmentada amb la informació de posicionament dels elements virtuals. Aquesta operació es pot realitzar de moltes maneres i solen ser enfocades a aplicacions d'usuari molt diferents. En el nostre projecte, plantejarem una **solució concreta basada en visió per computador que ens permetrà validar el nostre entorn** i complementar-lo per poder començar a pensar i experimentar amb aplicacions d'usuari. Es desenvoluparà aquesta solució específica d'RA amb l'ajuda de la llibreria de visió per computador *OpenCV* [19]. El procés de la solució (esquematzat a la figura 17) es compon de dues operatives principals: el **seguiment** i l'**alineació**. Les hem introduïdes en l'apartat d'antecedents i ara en farem la descripció del seu funcionament en una solució basada en visió per computador.

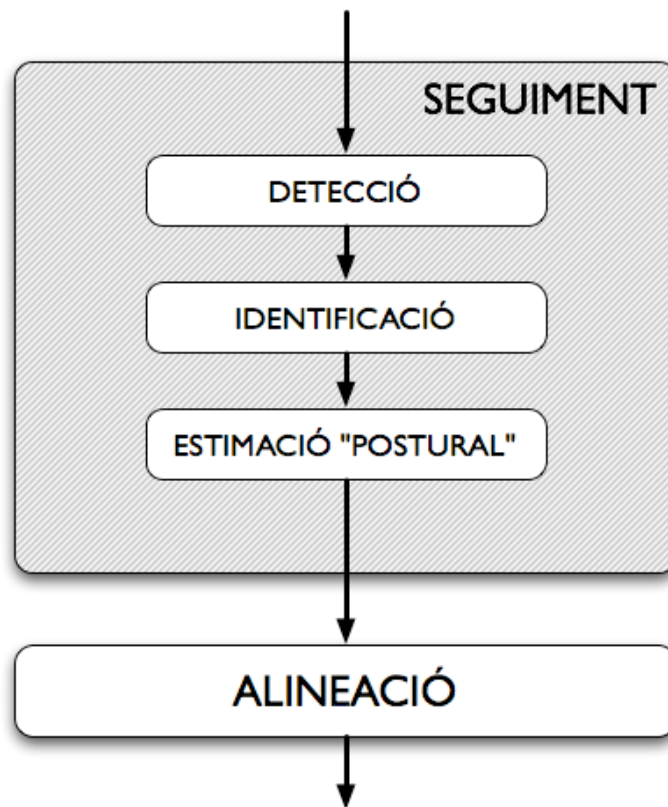


Figura 17: Etapes del processament de vídeo per la solució d'RA

3.5.1 Seguiment

El seguiment per la nostra solució es basa en la detecció de marcadors a l'escena. L'objectiu és que aquests marcadors ens facilitin la construcció d'un mètode robust i amb un alt rendiment per la **detecció i identificació**:

El **marcador**, aleshores, ha de ser un element que permeti facilitar-nos els primers passos que componen el seguiment: hem d'aconseguir minimitzar el còmput de detecció i identificació perquè és un procés que estarà avaluant contínuament el flux de vídeo. Això podria esdevenir un coll d'ampolla del sistema a l'hora d'obtenir una

bona representació en temps real. Els marcadors són elements que introduïm a l'escena de forma deliberada. Poden ser "tangibles", permetent així la interacció física amb l'usuari, o bé ser elements fixats sobre el món real, com a marques de navegació. Aquestes dues condicions ens permeten dissenyar un marcador amb unes característiques especials que ens ajudin a complir l'anterior requeriment: simplificar al màxim la seva detecció i identificació.

Basant-nos en el disseny d'altres marcadors existents de sistemes similars al nostre, com ara els ja mencionats al capítol 2, *ARtag* i *ARtoolkit*, avaluant també l'estudi de prestacions de diversos d'aquests marcadors de X. Zhang, S. Fronz i N. Navab a l'article *Visual Marker Detection and Decoding in AR Systems: A Comparative Study* [33], hem decidit començar amb un patró com el mostrat a la figura 18.

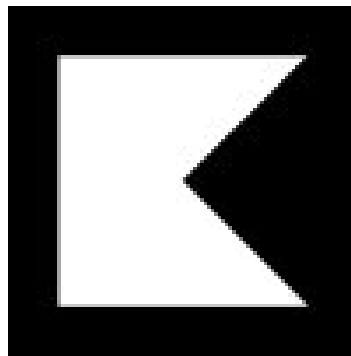


Figura 18: Patró marcador

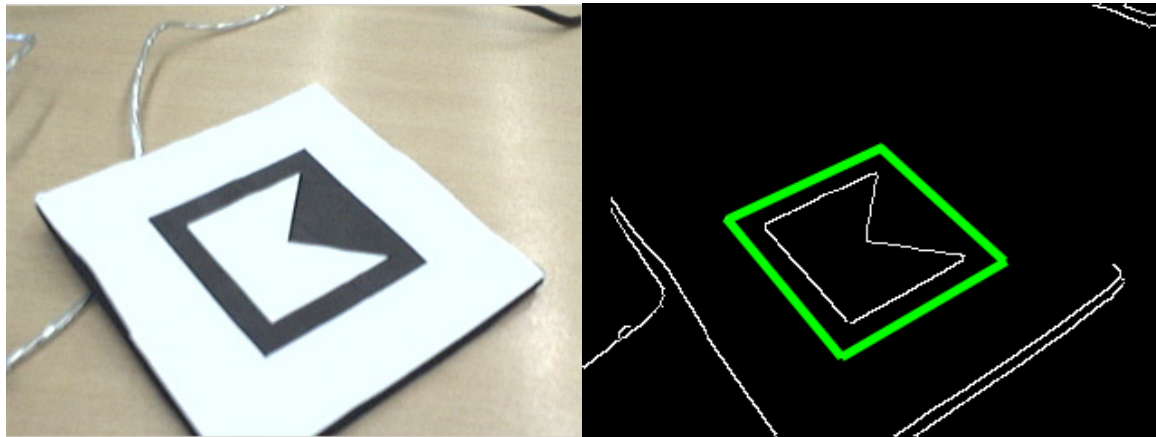
Aquest patró ens ha de facilitar la seva **detecció** gràcies a les seves definides fronteres (aquest alt contrast pot facilitar notablement deteccions de cantonades i contorns del patró). A més a més, aquest patró és difícil que es pugui trobar reproduït a les escenes reals sinó és perquè es situa en ella de forma expressa.

Amb la detecció encara som a un pas de conèixer si l'element detectat en l'anterior etapa és realment el nostre patró, per això procedim a la **identificació**. Aquest procés ens ha de permetre determinar si realment es tracta del patró cercat, i a més a més, obtindrem informació complementària sobre la posició que adopta el marcador a la imatge capturada. Per això es defineix una forma lliure de simetries al seu interior que pugui ser identificada de forma unívoca. Cal remarcar que aquesta etapa encara no pretén conèixer la seva posició al món real (o més estrictament parlant: la relació "postural" entre patró i càmera); sinó que només identifica la posició del patró marcador a la imatge capturada del vídeo a partir de les seves característiques.

Finalment, amb l'objectiu de millorar la resposta de la solució i l'obtenció de dades sobre el posicionament, s'apliquen altres algoritmes que van des de la predicció a la més simple estabilització. Es podrien plantejar al llarg del desenvolupament, ja que no tenen més repercussió que nous mòduls i "petites" modificacions del flux.

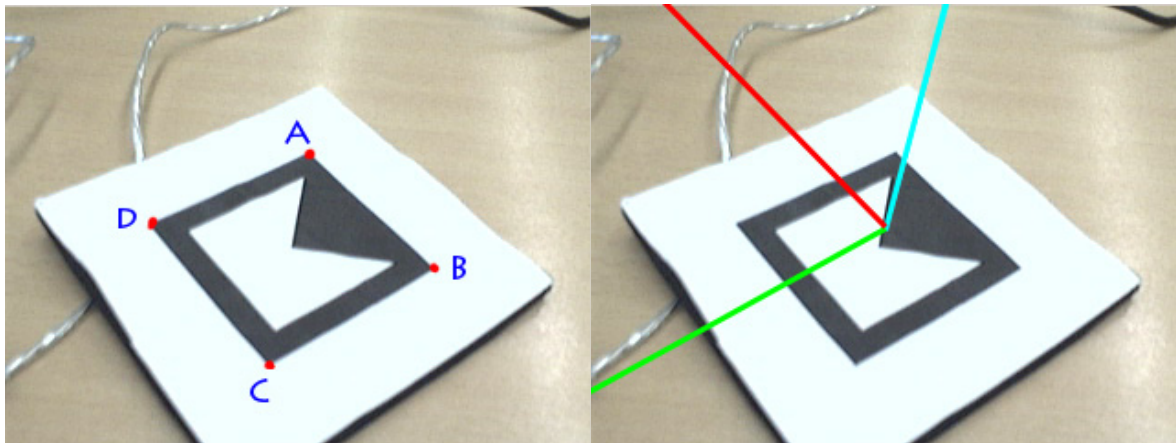
3.5.2 Alineació

Aquesta operació ha de permetre situar l'objecte en la seva disposició final. Es tracta d'un posicionament que afegeix a la seva postura ja calculada, la situació relativa de l'objecte en l'escena capturada i la seva mida d'acord amb el sistema de representació escollit. És una operació purament geomètrica de la qual ens en podem fer càrrec amb *C++* o fins i tot amb *OpenCV*.



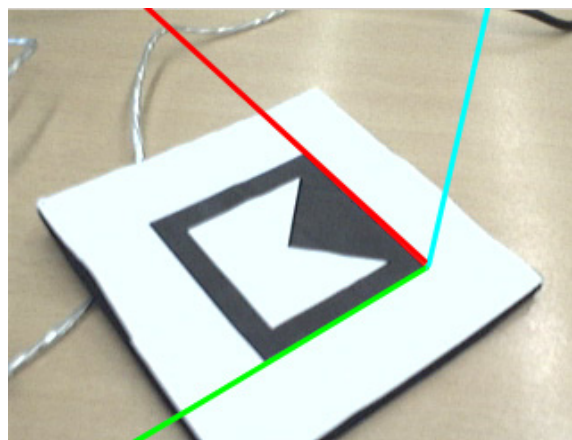
(a)

(b)



(c)

(d)



(e)

Figures 19a - 19e: Exemples de captura, detecció, identificació, estimació de postura i alineació.

3.6 Implementació de la solució d'RA

En aquest apartat tractarem els detalls d'implementació de la solució. Ho farem seguint l'ordre que en que s'apliquen a l'hora de processar el flux de vídeo i d'obtenir les dades:

1. Seguiment
 - a) Detecció: Pretractament de les imatges
 - b) Detecció: Localització de patrons a l'escena
 - c) Identificació del patró
2. Estimació de posició i alineació

3.6.1 Seguiment

a) Detecció: Pretractament de les imatges

El procés de detecció (figura 19b) s'inicia amb la necessitat de trobar patrons marcadors a l'escena capturada. Per això, el primer pas és adequar les captures per aconseguir identificar-los amb més facilitat.

El procediment triat s'inspira en els mètodes de treball d'altres autors com ara el de la llibreria *ArToolkit*, i la proposta de l'article *Robust 2D Tracking for Real-Time Augmented Reality* [34]. Ambdós proposen mètodes molt semblants on, després d'aplicar un filtre llindar per convertir la imatge en binària, fan una cerca d'àrees de punts connexes a fi d'eliminar les més petites, reduint així soroll en la imatge.

Per la nostra aplicació, després de fer diferents proves contemplant que l'entorn pogués treballar sobre diferents condicions d'il·luminació i que el patró pogués estar més proper o més llunyà (per tant, detectar-lo a diferents mides dins l'escena) el mètode que ha donat millors resultats es el següent:

En primer lloc, hem de contemplar que generalment les *webcams* estan dissenyades per treballar en condicions molt estables i els seus sensors no són especialment precisos en males condicions d'il·luminació. Tendeixen a generar molt soroll i fluctuacions en el color. És per millorar aquest problema de color, que la primera transformació a fer en la imatge de treball és ometre les components de color, convertint la imatge a escala de grisos.

Tot i així, la càmera segueix oferint una captura sorollosa i cal un mètode que pugui minvar aquesta problemàtica. Les primeres proves van passar per utilitzar una suavització gaussiana i l'aplicació d'algoritmes de morfologia binària com la combinació d'obertura i tancament [35]. Aquests mètodes eren prou bons per tractar posteriorment en una detecció de cantonades, però no en el mètode que apliquem definitivament de detecció de contorns.

La solució final està inspirada en un exemple disponible a la llibreria *OpenCV* [36] per la detecció de contorns en una imatge. Aquesta transformació consisteix a realitzar un *downsampling* i un posterior *upsampling* amb el mètode de piràmide gaussiana. Això és, de forma molt simplificada: reduir la imatge i tornar-la a ampliar.

A grans trets, l'operació de reducció es realitza eliminant files i columnes imparells i se'n millora el resultat amb un filtrat gaussià passa-baixes de la imatge reduïda. Per altra banda, l'operació d'ampliació insereix files i columnes zero en les posicions senars i en fa una convolució potenciada per crear la interpolació necessària. Així, veiem que el soroll es fa desaparèixer amb la suavització que implica el *downsampling* i a més a més aconseguim millorar les característiques de les cantonades amb l'*upsampling*. [37]

Finalment, el nostre objectiu és eliminar tota la informació que no és referent als contorns més marcats de la imatge on s'ha de trobar el nostre patró. Per això utilitzem una detecció de contorns basada en la implementació del mètode detector de contorns de *Canny* [38] de la llibreria *OpenCV*.

b) Detecció: Localització de patrons a l'escena

El disseny dels marcadors utilitzats es basa en tenir un alt contrast per ser contorns fàcilment detectables. Fins el moment només hem tractat la imatge per fer més evidents aquests contorns; ara cal triar quins dels contorns més rellevants de l'escena poden ser els dels patrons que cerquem a la imatge.

Cal notar que el desenvolupament d'aquest apartat comença guiat per l'ús de detecció de cantonades proposat per diversos treballs previs com el citat *Robust 2D Tracking for Real-Time Augmented Reality* [34] i el *survey Robust Feature Point Extraction and Tracking for Augmented Reality* [39].

La llibreria *OpenCV* compta amb la implementació de **detecció de cantonades** amb l'algoritme de *Harris* [40] i amb la **detecció de contorns**; vam avaluar ambdós per conèixer quin es podria adaptar millor a la nostra solució.

Amb el mètode de Harris, després de fer la detecció per trobar les cantonades més rellevants a l'escena, calia interpretar els espais convexos delimitats per les mateixes on trobar els possibles patrons de l'escena. Aquesta operació complica molt la lògica del procés, i en les proves realitzades comportava un còmput elevat, afectant així el rendiment de forma considerable.

Per altra banda, amb les funcionalitats d'*OpenCV* pel tractament de contorns es van poder obtenir resultats molt més eficients. L'estructura d'aquesta detecció és mostra a la figura 20.

Utilitzant la funcionalitat ***cvFindContours*** d'*OpenCV* podem obtenir tots els contorns que componen la nostra imatge en una estructura de dades. Com podem veure en el procés esquematitzat a la figura 20, analitzem totes les deteccions una a una aproximant-hi un polígon amb la funcionalitat ***cvApproxPoly*** i ens quedem aquells contorns formats per 4 segments, que a més formen un polígon convex (no han de ser quadrats, ja que si no s'observen des d'una posició totalment perpendicular es deformen). D'aquest subconjunt descartem els més petits (segons la mida de la diagonal més gran) que són deteccions sorolloses. Això ho fem sense afectar el rendiment de l'entorn: en el cas de que el patró sigui suficientment petit (és a dir, que estigui lluny) quedaria descartat per aquesta regla. No ens afecta ja que la càmera tampoc és capaç d'enfocar-lo i el mètode de detecció no en detecta els seus contorns.

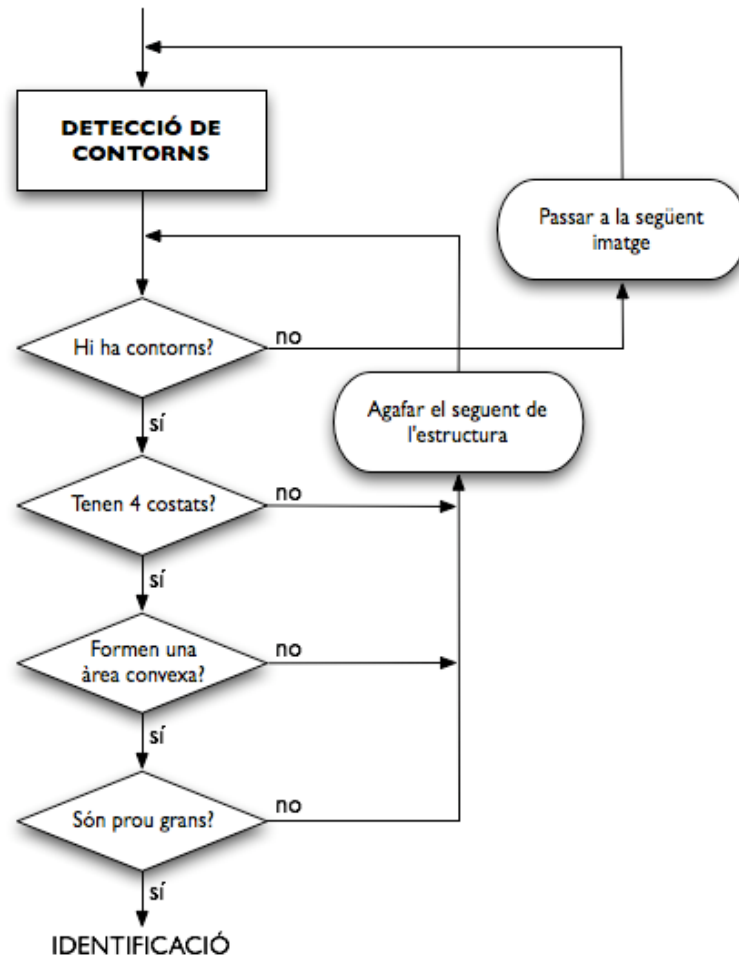


Figura 20: Flux de la detecció del patró

c) Identificació del patró

En el pas anterior hem detectat possibles candidats a patró marcador, però és evident que en una escena poden existir molts altres polígons rectangulars que podrien ser detectats en l'etapa anterior, així que hem d'aconseguir identificar (figura 19c) si cadascuna d'aquestes deteccions és realment o no un patró que introdueixi un nou element virtual.

Els contorns detectats es troben en una estructura de la qual podem recuperar els seus vèrtexs. Aquests ens serviran per tal de calcular una transformació geomètrica per situar-lo en una posició "origen" amb la qual pugui ser comparat (figura 21a). Aquesta transformació és una homografia i descriu la relació entre dos punts de vista del mateix pla (figura 21b). Es tracta d'una relació projectiva que ens permetrà mapejar el patró detectat en relació a un altre punt de vista. Aquesta relació és una matriu 3x3 que defineix una equació amb 8 graus de llibertat que es pot calcular entre 4 correspondències. En el nostre cas la calcularem entre els vèrtexs obtinguts i uns punts definits com origen: les posicions $\{x,y\} = \{0,0\}, \{100,0\}, \{100,100\}, \{0,100\}$. Aquesta operació d'obtenció de la transformació s'integra a la llibreria *OpenCV*, igual que s'hi integra l'aplicació de la transformació prèviament calculada a una imatge. Així podrem reproduir el contingut del contorn a identificar en l'àrea d'origen.

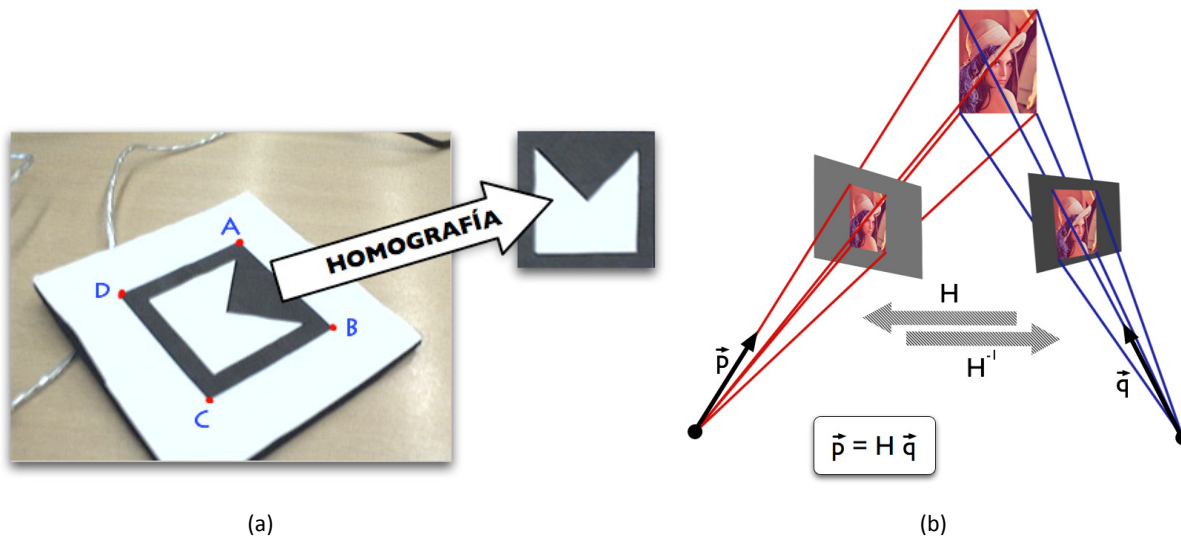


Figura 21: Homografia de la imatge capturada per reconstruir el contingut del patró detectat i descripció de la transformació projectiva

En les primeres implementacions teníem problemes amb els resultats d'aquesta operació, eren molt inestables; això era degut a que la funcionalitat de detecció de contorns no retorna una estructura de vèrtexs ordenada (i no estava documentat), així que abans de realitzar el càlcul de l'homografia s'han d'ordenar els vèrtex. Hem triat una la ordenació en el sentit de les agulles de rellotge com en els punts definits com a punts d'origen.

Un cop disposem de la reconstrucció, podem procedir amb la identificació de si aquest és el nostre patró. La metodologia triada per fer aquesta operació és una simple diferència entre patró original i el contingut del patró reconstruït a partir de la imatge. Això ens permetrà identificar si el patró és el que busquem i en quina posició està situat respecte als seus vèrtexs. A l'hora de fer aquesta identificació es podien aplicar molts altres algorismes de més precisió, com la també testejada correlació entre detecció i patró, però ens vam acollir a aquest mètode de diferència per tal d'assegurar un bon rendiment ja que implica un còmput molt més reduït.

Aquesta diferència és exactament una diferència en valor absolut (figura 22) amb la qual, al realitzar una resta entre la reconstrucció binaritzada i el model coincident, en resulta una imatge "negre": si en sumem tots els seus píxels la suma és molt propera a zero (a causa de les imperfeccions de la reconstrucció mai serà zero). En canvi, si la reconstrucció és diferent al model, el resultat serà molt diferent a zero. Per conèixer la seva posició, es compara amb el model en les 4 possibles rotacions; si alguna resulta propera a zero, identifiquem positivament el patró i en coneixem la seva posició. En canvi, si cap és propera a zero, el contorn detectat no pertany a un patró; no s'hi ha de reconstruir doncs, l'element virtual.

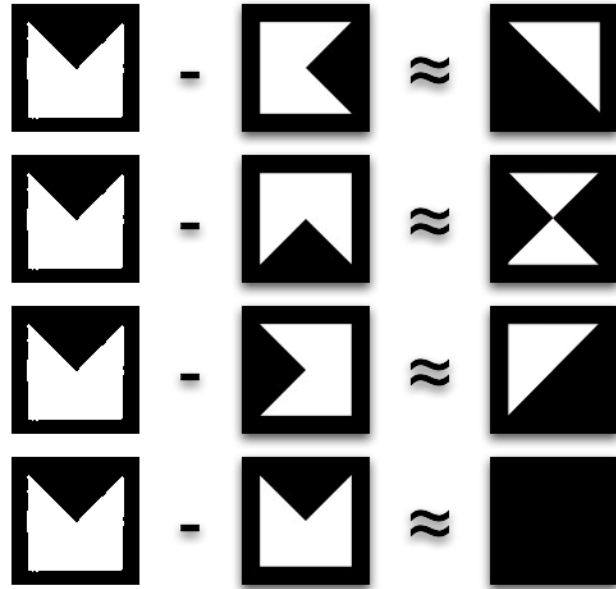


Figura 22: Procés d'identificació del patró

3.6.2 Estimació de posició i alineació

Arribats a aquest punt, hem recollit informació de posicionament d'un patró en una imatge. De fet, aquest seguit d'operacions s'aplica a un flux continu d'imatges capturat per la *webcam*, que està connectada a l'equip on desenvolupem el projecte. Per tal d'adequar-nos al requeriment d'una resposta en temps real, el procediment ha tingut en compte minimitzar tot tipus de costos. Això, sense oblidar en cap moment que aquesta simplificació no ha d'afectar el bon funcionament del sistema. Ara arriba l'hora d'interpretar les dades disponibles per tal d'extrapol·lar un sistema de referència tridimensional (figura 19d).

La nostra primera intenció era minimitzar els costos de còmput i evitar complicar l'entorn amb procediments molt complexos. Per fer-ho vam voler avaluar inicialment la possibilitat d'evitar mètodes de calibració envers mètodes que si que utilitzessin aquesta operació. A l'avaluar un mètode sense calibració assumíem que la distorsió de construcció de la lent i el sensor de captura d'imatges eren ideals, acceptant lògicament un cert rang d'error en la representació dels elements virtuals, donat que la lent està inevitablement subjecte a deformacions. De totes formes, podríem avaluar quin és aquest marge d'error.

Iniciàvem aquesta etapa doncs, basant-nos en mètodes explotats per diverses investigacions com el ja esmentat [34], a més a més de l'article *Augmented Reality Camera Tracking with Homographies* [41] o bé l'aproximació que fa [42]. Aquests mètodes basen l'estimació de posició de diferents maneres mitjançant homografies simples, com la que hem utilitzat per de reconstruir i comparar el contingut del marcador detectat.

Vam poder comprovar que aquestes homografies estaven d'alguna manera relacionades amb la matriu de visualització de l'espai tridimensional (figura 23). Copiant les dues primeres files de l'homografia inversa a la matriu de l'espai 3D i construint la tercera fila a partir del producte vectorial dels dos primers vectors, coincidia

amb la representació esperada (determinat mitjançant proves empíriques). El seu funcionament però, era molt rudimentari. A més a més, per un problema de treball en el sistema de coordenades d'*OpenCV* teníem encara més problemes per buscar una justificació i ajustar aquests resultats.

$$\begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & x \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$(g \quad h \quad i) = (a \quad b \quad c) \times (d \quad e \quad f)$$

$$x = ?$$

Figura 23: Reconstrucció de la matriu de visualització 3D a partir de l'homografia

Un cop descartats els mètodes sense calibració² per la seva mala precisió, vam explorar els principis de calibració i més concretament un dels mètodes més populars: el mètode Zhang [43]. A grans trets, aquest mètode pren un pla de referència el qual considera que es troba sobre el pla Z=0 amb el que redueix el problema a calcular una homografia entre el pla de projecció i el pla estimat. Aquesta homografia inclou els paràmetres intrínsecs i extrínsecs de la captura i n'extreu aquests paràmetres coneixent que les rotacions respecte al pla estimat són ortonormals i mètodes d'estimació matemàtics. En aquest mètode es pot entreveure que l'aproximació que havíem fet fins el moment era prou correcta, però que li faltaven molts detalls a tenir en compte per construir un mètode més complet i robust.

La calibració de la càmera podríem haver-la inclòs en el programa gràcies a les llibreries *OpenCV*, però vam creure més adient evitar aquest desenvolupament amb la utilització d'un software extern [44] que ens permetia obtenir totes les dades de calibració que necessitarem en la nostra solució: el punt central de la lent, la distància focal i els paràmetres de distorsió de la lent.

Disposant d'aquesta informació de calibració podíem aplicar algun algoritme implementat a *OpenCV* d'estimació de posició d'un objecte a partir de la seva representació en la captura. Vam començar per intentar aplicar *POSIT* [45] que era l'únic dels algoritmes que disposava de una documentació per aprendre a estructurar les dades necessàries per passar-li a la funció. Aquest algoritme però, no permet en principi l'estimació d'objectes plans; mentre que el nostre patró és pla. Existeix la possibilitat de modificar-lo per aquesta escomesa, però abans vam voler provar amb un altre mètode implementat a *OpenCV* i que coneixíem més, el mètode d'estimació que inclou Zhang en el seu mètode de calibració. Es tracta d'un mètode basat en la solució d'un sistema d'equacions lineals amb *SVD* (*Singular Value Decomposition*) [46].

En aquest cas, després d'adequar les dades a les estructures requerides per la funció vam poder disposar de

² Cal notar que hi ha alguns mètodes moderns que presumeixen de ser mètodes sense calibració, però es tracten més aviat de sistemes que realitzen una calibració online al llarg de la seva execució.

l'estimació "postural" sense problemes. La sortida d'aquesta funció expressa la seva estimació en un vector de 3 posicions que indica rotació en els tres eixos de l'espai tridimensional.

Cal notar que amb el mètode Zhang no es pot estimar més que la postura del objecte; no pot estimar per tant la translació del punt de vista per situar l'objecte a la seva posició final. Es tracta d'un problema matemàtic sense solució. Per fer l'alineació (figura 19e) s'hauria de disposar de més vistes de l'objecte (coneixent la relació espacial entre les vistes) i solucionar-ho per exemple mitjançant mètodes de triangulació. Així que, en principi no tindriem solucionat l'apartat de l'alineació del model virtual.

La llibreria *OpenCV* tanmateix, contempla aquesta estimació en la mateixa rutina. La solució implementada l'havíem trobada en diverses propostes: es tracta d'utilitzar l'anomenada "perspectiva dèbil" (*weak perspective*) per adaptar el model.

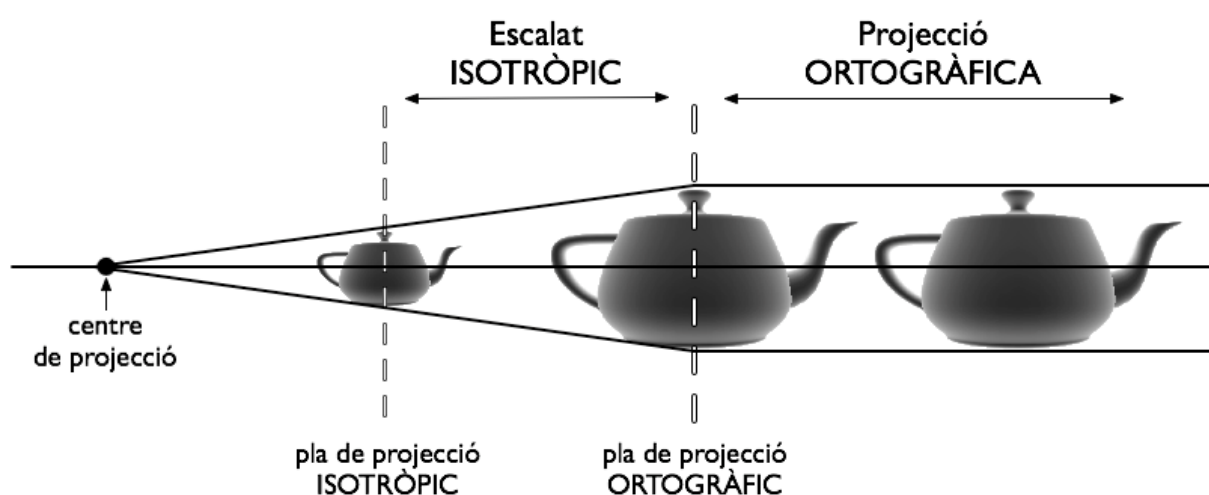


Figura 24: Model de perspectiva dèbil

La *weak perspective* (figura 24) es basa en treballar amb un objecte transformat amb una projecció ortogràfica a una distància coneguda per després fer-ne un escalat isotròpic (és a dir, sense emmirallat). Amb aquest mètode s'aconsegueix afegir la informació necessària de profunditat per calcular un resultat aproximat d'alineació amb un marge d'error gairebé imperceptible (almenys en les nostres proves).

Amb això ja disposem de totes les dades de posicionament i postura necessàries per enviar a l'entorn i que faci la representació final de l'escena augmentada. Però aquestes dades les hem d'adequar per enviar-les a l'entorn.

En primer lloc, hem de traduir la representació de rotacions obtinguda amb l'algoritme d'estimació "postural" per incloure'l en l'espai tridimensional creat en *OpenGL*. Hem de construir una matriu 4x4 amb els paràmetres extrínsecs que aplicarem a la visualització d'*OpenGL* per tal de posicionar el punt de vista i poder representar el món virtual com ens indica el patró marcador identificat. La representació de rotacions és coneguda com la fórmula de rotació de *Rodrigues* [47] i la podem traduir a matriu amb la llibreria *OpenCV* (figura 25). És una operació relativament senzilla basada en càlculs trigonomètrics.

$$r_{rodrigues} = (r_x \ r_y \ r_z)$$

$$\varphi = \|r_{rodrigues}\|$$

$$r_{rodrigues} = \frac{r_{rodrigues}}{\varphi}$$

$$M_{\text{visualització}} = \cos(\varphi)I + (1 - \cos(\varphi))rr^T + \sin(\varphi) \begin{pmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ r_y & r_x & 0 \end{pmatrix}$$

Figura 25: Generació de la matriu de visualització per a OpenGL

Finalment, per tal d'adaptar correctament l'espai tridimensional a la "deformada" captura de la realitat, hem de construir una matriu 4x4 de projecció que tingui en compte els paràmetres calculats en el mètode de calibració. Aquesta projecció es construeix com podem veure a la figura 26 (mètode basat en l'utilitzat per la llibreria *ArToolkit* [14]).

$$\begin{pmatrix} 2 * f_x/w & 0 & (2 * c_x/w) - 1 & 0 \\ 0 & 2 * f_y/h & (2 * c_y/h) - 1 & 0 \\ 0 & 0 & (fp + np)/(fp - np) & -2 * fp * np/(fp - np) \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

On f_x i f_y són les distàncies focals,

c_x i c_y el centre real de la imatge,

fp i np els plans frontal i posterior del volum de vista,

w i h l'ample i alt de la projecció

Figura 26: Generació de la matriu de projecció per a OpenGL

A l'hora de fer el dibuixat final a *OpenGL* aquest es refereix a dues matrius que li estableixen els paràmetres anteriors (visualització a la matriu *MODELVIEW* i projecció a la matriu *PROJECTION*). Per tant, l'operació d'especificació d'aquestes dades a *OpenGL* s'aconsegueix amb la càrrega de les matrius calculades per *OpenCV* mitjançant la primitiva ***glLoadMatrix*** en cadascun dels contextos.

Capítol 4

Anàlisi de resultats

Al llarg del capítol anterior hem vist l'estudi i la implementació de l'entorn de desenvolupament de Realitat Augmentada plantejat per a aquest projecte, a més a més d'una solució específica preparada per tal d'avaluar el funcionament d'aquest entorn. Un cop vistos aquests detalls, volem fer una anàlisi del seu funcionament per poder veure amb ull crític els punts forts i febles de la nostra implementació. Analitzarem el resultats relatius a l'entorn, a la solució i també referents als resultats del disseny i implementació.

4.1 Resultats de l'entorn

L'entorn és en certa manera l'esquelet del nostre projecte tot i no ser l'encarregat d'executar les tasques més complexes i de més requeriments computacionals. Però, per ser operatiu, ha estat subjecte a restriccions importants de rendiment. Una de les principals restriccions que imposa el treballar amb Realitat Augmentada és la seva necessitat, per definició, de ser una tecnologia que treballa en temps real. S'han de poder construir elements virtuals sobre la realitat de forma contínua i perfectament coordinada. Només així es pot presentar uns bons resultats útils per construir aplicacions d'usuari. La restricció que suposa operar en temps real afecta a totes les parts del desenvolupament d'aquest projecte, fins i tot l'entorn. Per això han de poder obtenir un bon *throughput* donada la seva col·laboració en la generació d'RA.

L'operativa de l'entorn comença en la **captura de vídeo** per poder presentar la part real de la mescla amb els objectes virtuals; una operació que no ha suposat cap problema per posar-la en marxa amb un rendiment acceptable. Com comentàvem, el *hardware* de captura permet obtenir un rati de 30 imatges per segon que és suficient per presentar aquest flux en pantalla sense problemes.

El problema principal de rendiment en l'entorn recau en la **mescla d'imatges tridimensionals i la captura de vídeo**. Per enfrontar aquest problema, en el desenvolupament s'han dut a terme proves de rendiment sobre dues implementacions diferents que permetien fer aquesta operació. El resultat ha desmarcat una d'elles, que ens ofereix una representació tridimensional superposada al vídeo a un rati superior a les 60 imatges per segon (es dupliquen les mostres de vídeo a base de repetir-les).

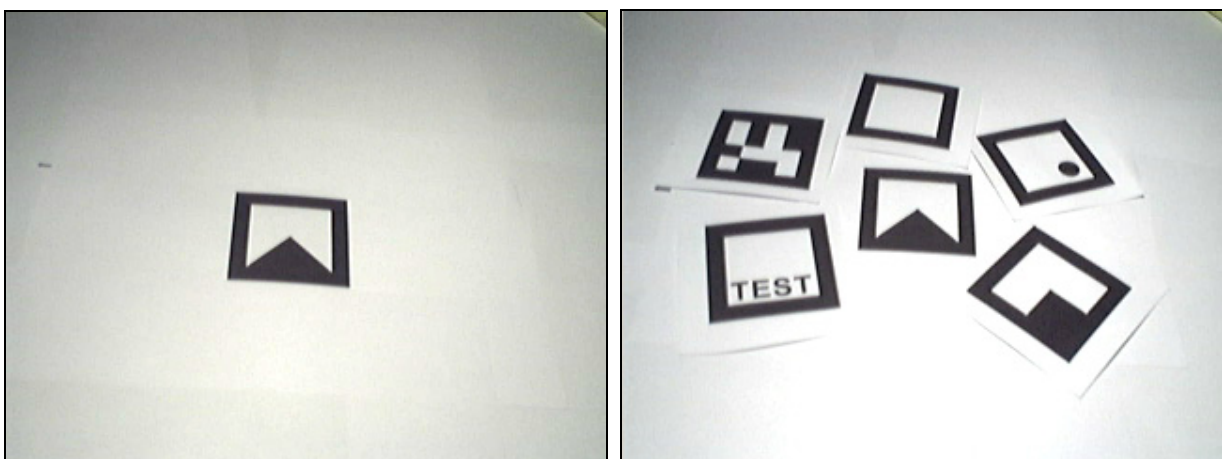
4.2 Resultats de la solució

El que entenem per solució en el nostre projecte és l'apartat del desenvolupament que realitza les tasques d'estimació de posicionament pels elements virtuals respecte a la situació de la captura de vídeo sobre el món real. La solució realitza aquesta operació mitjançant un processament del vídeo amb tècniques de tractament d'imatges i visió per computador per detectar patrons marcadors a situats a l'entorn que es captura. Aquests marcadors tenen la finalitat d'indicar els orígens de situació d'elements virtuals. És el que es sol classificar com una interfície tangible basada en marcadors.

Aquesta operació, igual que en el pas anterior, ha de respondre a un ritme suficient per poder estimar la posició dels patrons a cada pas per després aplicar-la als elements tridimensionals i que passin a "formar part del món real" alineats al la posició esperada en la captura de vídeo. D'aquest procés, a més a més d'un bon rendiment, s'espera que sigui el més exacte possible per obtenir una representació realista. Analtzarem el comportament de la nostra implementació respecte aquestes característiques: el rendiment de l'augmentació i la seva robustesa.

a) Rendiment de l'augmentació

Volem analitzar el rendiment d'augmentació per avaluar el **temps de resposta de l'estimació de postura i posicionament** un cop detectat i identificat el patró marcador (funcions que estudiarem en el següent apartat). Això implica tenir en compte el consum temporal des de l'estimació fins a la representació del objecte virtual acord amb l'estimació. Hem realitzat aquest càlcul temporal incloent a la nostra solució registres de temps just abans de començar el procés i just després d'acabar-lo amb la funcionalitat de l'API de Windows GetSystemTime. Notar que, donat que aquest procés pot fluctuar per errors en la detecció i identificació del patró, presentem el resultat com una mitjana.



(a)

(b)

Figura 27. Mostres de captures utilitzades en els tests

Així doncs, hem pogut calcular que el procés triga una mitjana de 8 milisegons en realitzar l'estimació i adaptar l'espai tridimensional a aquesta. Afortunadament, aquesta operació és molt eficient i no suposarà un coll d'ampolla en el nostre sistema. La prova es realitza sobre una mostra de 100 imatges com la representada a la

figura 27a. Es tracta d'una captura de 320x240 píxels d'un patró de 7cm de costat. Es realitza a 40cm del patró marcador amb un angle de 45 graus i una il·luminació difosa sobre el patró. En podem veure un esquema del muntatge representat a la figura 28.

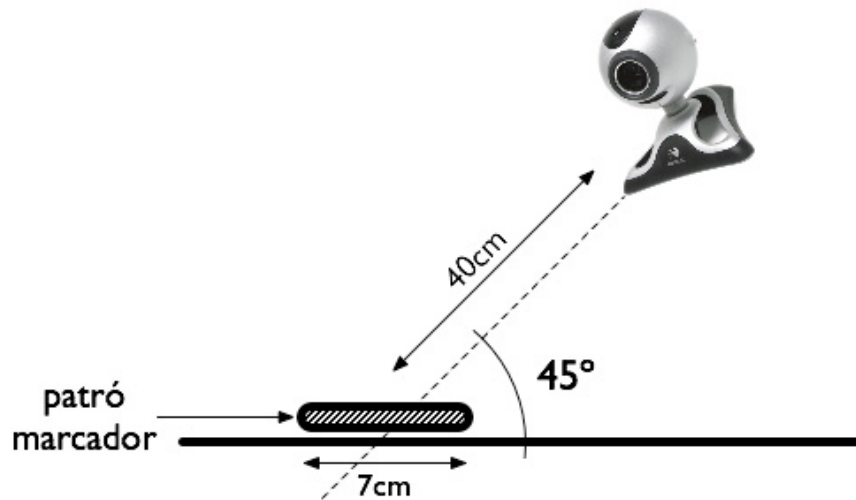


Figura 28. Muntatge de la captura d'imatges per les proves

b) Robustesa de l'augmentació

Com comentàvem, la solució implementada es basa en un sistema de detecció amb visió per ordinador de marcadors. El marcador utilitzat per a les proves es basa en una simplificació del mateix concepte que presenten solucions similars utilitzades en llibreries d'RA com ArToolkit [14], ArTag [15] i d'altres. Aquest disseny es basa en un estudi [33] que en fa una comparativa de diversos ja existents, i per això creiem desitjable comparar els resultats obtinguts en la nostra aproximació amb l'estudi d'aquests patrons. No es tracta d'obtenir un *ranking* envers aquests, sinó d'avaluar els seus punts forts i febles.

- El primer dels passos a avaluar és el **cost "net" de la detecció i identificació del patró a l'escena**. És a dir, el temps que triga a fer el pretractament, detecció de contorns i identificació del patró en l'escena en una condició favorable per fer aquest procés. Per fer-ho hem escollit treballar en un entorn com el representat a la figura 27a: una il·luminació difosa (que millora l'estabilitat de recepció de la càmera) i un context d'escena buit (per no treballar amb la complexitat que comportaria un entorn més heterogeni). En aquestes condicions, hem calculat el temps mitjà que suposa realitzar tot aquest procés: en total, 26'84 milisegons. Es tracta d'un càlcul fet, novament, sobre una mostra de 100 frames. El seu comportament fluctua per la captura sorollosa de la càmera entre 15 i 32 milisegons. Aquest comportament és lleugerament més lent a les mitjanes obtingudes en mètodes comparats a l'article de referència, on ronden entre 4'10 i 11'60 milisegons en una detecció similar. Això és degut a que, probablement, aquestes implementacions són específiques pel problema que solucionen i en el nostre cas ens basem en un mètode construït amb operatives d'una llibreria que no implementa una solució totalment ajustada pel nostre problema concret.

- El segon test avaluat és el **cost “brut” de la detecció i identificació del patró a l’escena**. En aquest cas s’avalua el seu comportament en un entorn amb 5 patrons diferents al cercat, el que hauria de complicar-ne la seva detecció. En unes condicions novament favorables (podem veure’n un exemple a la figura 27b) s’avalua el seu temps de resposta: 77’09 milisegons. La prova ha estat realitzada novament sobre una mostra de 100 frames per alleugerir l’efecte del soroll en la captura de la càmera. Aquesta detecció és, com hem predit, més complexa en temps. De fet, no és una relació directament proporcional però parlem de gairebé un 300% d’augment al passar d’1 a 6 patrons a l’escena. Però com podem veure a l’estudi [33], aquest és un problema que compartim amb d’altres sistemes com per exemple el d’ArToolkit. S’ha d’afegir que en una prova igual amb una mostra major (de 500 imatges) només hi ha hagut un error d’identificació en un dels patrons, donant per bo el patró que conté la paraula “TEST”.
- Encara queda un tercer test, el **cost “natural” de la detecció i identificació del patró a l’escena**. Ara el patró es situa en un entorn heterogeni i s’avalua novament el temps de resposta d’aquest procés. Podem veure una de les 100 imatges utilitzades per la prova a la figura 29. En aquest cas, la detecció de contorns treballa amb més càrrega al detectar molts més candidats a patró marcador, el que implica alhora més processos d’identificació. Però veiem que en aquest cas la complexitat temporal de l’algorisme no puja tant com a la prova anterior: en aquest cas parlem d’una mitjana de 31,84 milisegons per la detecció i identificació positiva del nostre marcador.



Figura 29. Muntatge de la captura d’imatges per la prova de detecció i identificació en un entorn heterogeni

Cal remarcar que les proves de [33] i les nostres són realitzades en equips diferents, càmeres diferents i condicions d’iluminació diferents, tot i que també hem intentat ajustar les nostres proves a un entorn plenament favorable per la detecció i identificació. Aquests paràmetres poden tenir molt a veure en els temps calculats, sobretot si ens centrem amb la captura de la nostra càmera que té deficiències (com l’apreciable descompensació lumínica radial i petites fluctuacions en la lluminositat i enfocament al llarg de la captura). Tot i així, els resultats han estat validats en diverses repeticions i els resultats obtinguts són molt propers. Reforcem així que podem considerar aquestes mostres com una bona mesura orientativa per caracteritzar l’entorn.

Els tests realitzats fins al moment indiquen que tot i que el seu rendiment és més baix respecte d'altres sistemes existents, en unes bones condicions (les quals impliquen una bona il·luminació i evitar disposar molts patrons a l'escena) la identificació del patró en la nostra solució ofereix un resultat suficient per fer augmentacions a més de 30 imatges per segon (30 fps = 33'33 milisegons per el processament en la imatge).

Un cop vist això, passarem a l'avaluació de la **recognoscibilitat del patró**. En aquest cas avaluarem en quin rang és detectable (tant en angle com en distància). Per realitzar aquesta operació cal recórrer novament a un model favorable. En aquest cas és important tenir en compte que segons el material amb el qual es construeix el patró i els punts de llum pot ampliar-se el marge d'error per reflexions que dificultin la captura. En el nostre cas, hem evitat aquest tipus de problema des de bon començament dotant l'escena d'una il·luminació difusa i un material no reflectant. Hem obtingut els següents resultats:

- La **distància de detecció** és el primer aspecte morfològic del punt de vista a analitzar. Es tracta de reconèixer com afecta la mida de la captura a la detecció del patró. Les proves realitzades ens mostren que el patró de 7cm de costat es pot identificar des d'una distància mínima de 13cm, on ocupa el màxim rang de pantalla (figura 30a), i una distància màxima de 116cm, on deixa de ser identificat (figura 30b). Això indica que el patró es pot identificar amb una mida en pantalla d'aproximadament 20x20 píxels. Un resultat molt proper al millor dels presentats a l'article [33], que aconsegueix una descodificació a 14x14 píxels. Pel que fa a la detecció de mida màxima podem observar que pot ser realitzada fins i tot amb una distorsió d'enfocament considerable.

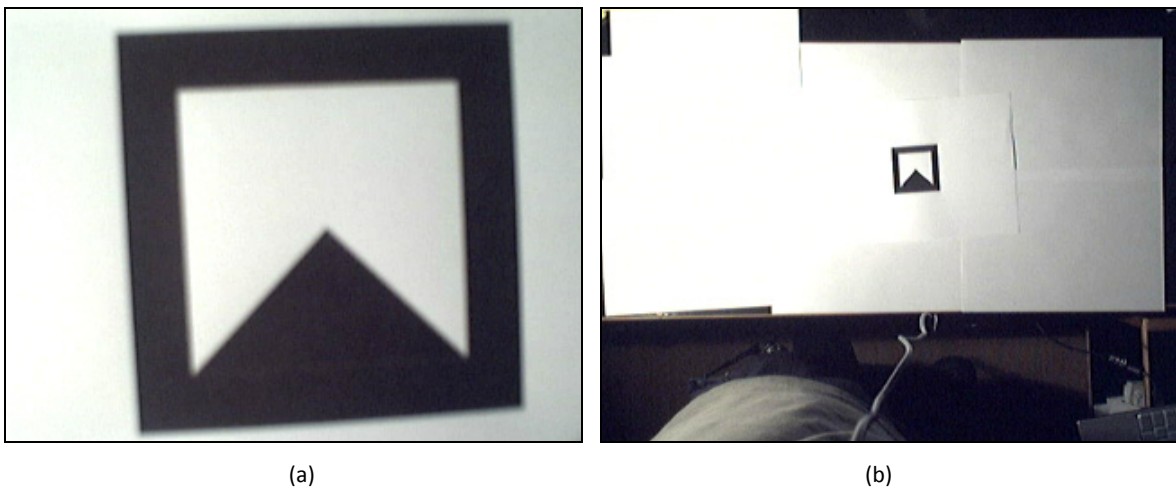


Figura 30. Mostres de les captures utilitzades en els tests de distància de detecció

- El següent aspecte morfològic que ens permet caracteritzar la nostra solució és l'**angle màxim de detecció i identificació**. Es tracta d'una prova realitzada desplaçant la càmera al voltant del marcador i anotant-ne l'angle que marca el llindar entre la seva detecció i identificació o no. Utilitzant un muntatge com el de la figura 31, girem la càmera situada a 40cm del patró. Dirigit-nos de 90 a 0 graus, podem observar que la detecció i identificació del patró degenera considerablement a partir dels 20 graus i deixa de ser detectat sobre els 15.

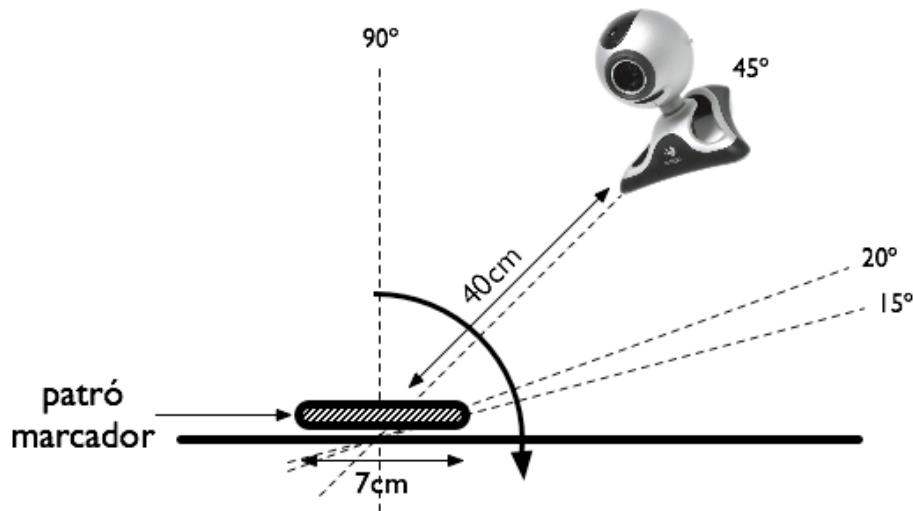


Figura 31: Muntatge de la captura d'imatges per les proves d'angle

4.3 Resultats del disseny i la implementació

Finalment, volem posar sota una anàlisi crítica els resultats finals de la nostra implementació envers al disseny inicial. Es tracta de veure en què s'han acomplert les expectatives relatives a la implementació i què caldria millorar pel seu ús en futures implementacions.

El disseny ha evolucionat al llarg de tot el projecte per adaptar-se a les diverses necessitats que hem trobat a l'hora de fer tests amb l'algoritme implementat en la nostra solució específica de Realitat Augmentada. Aquests canvis en la implementació han portat problemes en alguna ocasió. Ens referim, sobretot, a la separació conceptual entre entorn i solució. En la última etapa del projecte s'ha vist afectada la plena independència d'aquests mòduls al fer proves d'implementació per tal de permetre la detecció de múltiples patrons i marcadors orientats a disposar elements virtuals independents a l'escena. Tot i que és possible realitzar aquesta operació de forma independent no s'ha pogut finalitzar amb el temps i recursos destinats al projecte de final de carrera. Es tracta d'una operació que ha de tenir en compte molts més paràmetres, com per exemple, que si en una captura es detecta diverses vegades un candidat a patró, s'ha d'estudiar quin d'aquests resultats és més robust (en cas de tractar-se del mateix patró) o a quin d'ells correspon cadascuna de les identificacions (en el cas de tractar amb múltiples marcadors).

Un altre aspecte que volem destacar és el mètode de captura de vídeo. La implementació va ser iniciada amb la llibreria DirectShow per accelerar el temps d'inici de la implementació. Era necessari fer-ho per fer proves de concepte i avaluar el correcte funcionament de la mescla entre 3D i vídeo per continuar amb la resta del projecte o cercar alternatives en cas d'obtenir mals resultats. Però la idea també era arribar a passar aquesta captura a un mètode basat en la llibreria OpenCV (que sobre Windows també s'implementa amb DirectShow) que permetria passar l'entorn més senzillament a altres plataformes com GNU Linux sense canvis substancials en el mateix. Es va

arribar a implementar però en el temps dedicat a aquesta tasca no es van obtenir resultats operatius sobre Windows, tot i que sí que es van obtenir sobre GNU Linux. Seria desitjable aconseguir fer operativa aquesta implementació per treballar amb més llibertat en futures utilitzacions de l'entorn.

Pel que es refereix a la resta del disseny i implementació, tot ha estat realitzat satisfactòriament: s'han complert els objectius que permeten a l'entorn i la solució generar Realitat Augmentada, així com aconseguir que el seu comportament sigui en temps real.

Capítol 5

Conclusions

Després de passar per totes les fases del projecte (estudi, anàlisi, disseny implementació i proves) dedicarem aquest últim capítol de la memòria a realitzar una visió general del mateix i fer-ne una avaluació dels resultats obtinguts. A més a més, també complementarem aquest resum amb el nostre punt de vista sobre quines poden ser les possibles millores i futures ampliacions que se'n podrien derivar.

5.1 Visió general del projecte

A l'inici d'aquest projecte ens proposàvem fer un entorn de desenvolupament de Realitat Augmentada. Al llarg del projecte s'han aconseguit, però, més objectius: per una banda, s'ha fet una **revisió completa i actualitzada de l'estat de l'art de l'RA** i s'ha fet una **descripció del funcionament genèric d'un entorn tipus d'RA**. Pel que fa als objectius inicials, podem dir que s'han complert els previstos: s'ha realitzat el **disseny i implementació d'un entorn i una solució d'RA**, que pot servir d'entorn de proves per a futures investigacions. A més a més, el rendiment d'aquest és prou correcte per fer proves i prototips.

La **revisió de l'estat de l'art** en la investigació d'aquesta tecnologia s'ha presentat mitjançant un recull de les característiques a les quals pot estar restringit un entorn de Realitat Augmentada, orientat també a emmarcar el projecte segons les nostres possibilitats. Aquesta anàlisi complementa anteriors estudis de l'estat de l'art com [4][5] (dels anys 1997 i 2001 respectivament) amb noves aportacions actualitzades, tenint en compte aplicacions més modernes i avenços realitzats en la investigació d'aquest àmbit. Recopila doncs, tot un seguit d'investigacions i informacions que es trobaven fins ara disperses; almenys en el nivell d'actualització que aquí s'han presentat.

L'estudi ha servit també introduir una **descripció del funcionament genèric del d'un entorn d'RA** per tal d'obtenir els criteris necessaris per dissenyar el nostre propi entorn. Així, hem vist que els requeriments imprescindibles i els desitjables. A partir d'aquestes indicacions, i les pròpies derivacions de la creació d'un entorn de desenvolupament, es va **dissenyar** el projecte tot diferenciant-ne dues parts: per una banda, *un entorn* per realitzar les tasques referents a la mescla d'objectes virtuals i realitat i per altra banda, *una solució* específica basada en visió per computador que el complementa implementant un algoritme d'estimació de posicionament basat en marcadors.

Un cop descrit aquest disseny hem continuat amb la seva **implementació** en basant-nos en un desenvolupament en C++ amb l'ajuda de les llibreries gràfiques OpenGL i OpenCV pels algoritmes de visió per computador i

tractament d'imatges requerits per la solució. Per acabar aquesta etapa hem recollit en un **anàlisi de resultats** les proves que caracteritzen el funcionament del nostre entorn .

En aquesta anàlisi hem posat a prova el comportament del nostre entorn, estudiant-ne el seu rendiment. També hem estudiat el rendiment i la robustesa de la solució proposada, mitjançant proves de comportament en diferents escenaris que han permès avaluar els seus punts forts i febles que, a la vegada, hem comparat amb d'altres sistemes existents de Realitat Augmentada.

Pel que fa a la planificació temporal, podem dir que aquesta ha estat seguida sense incidències destacables. Tot i així, com és lògic, les etapes descrites eren força genèriques, de manera que a la pràctica no esdevenen tant acotades en el temps. Ens referim sobretot a la relació entre l'anàlisi, disseny i desenvolupament; etapes durant les quals sovint s'ha hagut de treballar en paral·lel per obtenir millors criteris i resultats.

5.2 Millores, ampliacions i futur del projecte

Millores al projecte

El projecte podria ser millorat en quatre línies principals: la depuració de la independència entre entorn i solució, l'optimització dels rendiments de la solució, la inclusió de la funcionalitat de detecció de múltiples patrons i l'afinament d'estimació de posició de l'entorn.

- Per una banda, seria molt recomanable **millorar la independència entre entorn i solució**, ja que com hem comentat al capítol d'anàlisi de resultats, durant el procés de desenvolupament del projecte va quedar minvada a causa de diverses proves d'implementació.
- Per altra banda, seria convenient **optimitzar els rendiments de la solució**, ja que aquests resulten per sobre de la mitjana d'investigacions similars. Per fer-ho, es poden avaluar altres mètodes de còmput inclosos a la llibreria OpenCV o, fins i tot, prescindir d'aquesta llibreria i implementar-ho manualment..
- També seria molt desitjable incloure en l'algorítmica de la solució la **detecció de múltiples patrons** que permetin l'augmentació de diversos espais virtuals alhora. Hi ha algunes proves ja iniciades, però obtenir un bon resultat requereix més temps d'implementació.
- Finalment la detecció podria comptar amb algoritmes complexos que permetessin **afinar l'estimació de la posició del patró**. Aquestes propostes van des de la inclusió de mètodes de predicció de moviment a estabilitzacions de la recepció de dades.

Ampliacions de l'entorn

A més a més d'aquestes millores, i fora dels recursos disponibles pel projecte, serien destacables les següents ampliacions de l'entorn:

- Incloure en el disseny de l'entorn suport per a processos interactius. Això tindria per objectiu permetre fer proves d'implementació ràpida d'aplicacions sense recórrer a utilitzar l'entorn com a "llibreria".

- Permetre l'ús de models virtuals més complexos i que puguin, per exemple, exportar-se des de bases de dades o fitxers d'entorns CAD de disseny 3D (Blender, Maya, 3DStudio, etc.).
- Es pot considerar també la inclusió de mètodes que permetin obtenir una representació més realista dels elements virtuals sobre la realitat, com pot ser la contextualització d'ombres i reflexions.
- Una altra de les branques amb possibilitats d'expansió en l'entorn està en l'ús de múltiples punts de vista per ampliar les possibilitats dels algoritmes de visió per computador i també per el seu ús en displays estereoscòpics.
- Una forma d'optimitzar l'entorn i la solució podria ser també la implementació d'algoritmes basats en l'ús de processadors gràfics (*GPU*) disponibles en el hardware de l'equip.
- Finalment, com ja hem comentat al llarg de la memòria, l'entorn pot ser fàcilment portable a altres sistemes operatius (fins i tot convertint-lo en un sistema multi-plataforma) utilitzant la llibreria OpenCV per la captura de vídeo i prescindint d'MFC de Windows per passar a un entorn GLUT, o bé treballar amb algun sistema de finestres multiplataforma com pot ser la llibreria QT o GTK.

Propostes d'aplicació futures

El nostre projecte permet la llibertat de desenvolupar proves de concepte en l'apartat dels algoritmes d'estimació que requereix un entorn de Realitat Augmentada. Però també permet la implementació d'aplicacions i no volem acabar la nostra anàlisi sense veure quines poden ser les opcions de desenvolupament futures:

- S'obre la porta doncs, a dissenyar en primera instància noves solucions d'estimació. Ens referim sobretot a solucions basades en mètodes de visió per computador i, més concretament, a la implementació de mètodes basats en el model del punts de referència no explícits; mètodes que s'aprofitin de característiques disponibles a l'entorn i no s'hi hagin d'incloure patrons per realitzar el còmput. Aquest és un dels àmbits en ple auge investigador en el camp de la Realitat Augmentada.
- També podem proposar el disseny de tot tipus d'aplicacions de la Realitat Augmentada, com per exemple, en l'àmbit de la imatge mèdica, aconseguint mètodes que permetin fer tracking del cos del pacient i fer augmentació d'imatges del seu interior obtingudes per radiofreqüència. Com veiem a la introducció d'aquesta memòria, es tracta de continuar investigacions com les referenciades a [7][8][9].
- De la mateixa manera, podem pensar en infinitat d'aplicacions que utilitzin la realitat augmentada com a interfície amb ordinadors personals en el disseny assistit per ordinador, en ajuda a la manufactura industrial, en navegació, jocs, multimèdia...
- La Realitat Augmentada evoluciona també en la seva variant amb l'ús de sensors i geolocalització per tal de fer les estimacions pertinents. Explorar aquest camp amb el nostre entorn podria ser de gran interès.

5.3 Valoració personal

Vull acabar aquesta memòria remarcant que realitzar aquest projecte ha estat una tasca molt enriquidora en molts sentits. Primerament, ha estat molt engrescador poder entrar a fons en aquest món en el que tenia posat molt d'interès. També ho ha estat el poder enfrontar d'una forma autònoma totes les etapes del projecte i solucionar-ne les dificultats plantejades al llarg del mateix. Vull destacar també que la Realitat Augmentada és un camp apassionant que m'ha donat la possibilitat d'unir conceptes de dues disciplines com són els gràfics 3d i la visió per computador. Això m'ha permès, lògicament, aprofundir en ambdues disciplines alhora, es aspectes com ara la visualització, la calibració de càmera, la detecció de característiques, etc. En definitiva, he de dir que el projecte ha estat una experiència molt profitosa, amb l'àmbit de la qual espero poder continuar treballant en un futur proper.

Bibliografia i referències

- [1] English Wikipedia, "Ivan E Sutherland", May 2008. [Online]. Available: http://en.wikipedia.org/wiki/Ivan_Sutherland
- [2] I. E. Sutherland, "The ultimate display", Proceedings of IFIP Congress, vol. 2, pp. 506-508, 1965.
- [3] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays", IEICE transactions on information and systems, vol. E77-D, no. 2, pp. 1321-1329, 1994.
- [4] R. T. Azuma, "A survey of augmented reality", Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355-385, August 1997.
- [5] R. Azuma, Y. Bailiot, R. Behringer, S. Feiner, S. Julier, and B. Macintyre, "Recent advances in augmented reality", Computer Graphics and Applications, IEEE, vol. 21, no. 6, pp. 34-47, 2001.
- [6] R. Spring and M. Braun, "Enkin for android" [Online]. Available: <http://www.enkin.net/Enkin.pdf>
- [7] A. State, M. A. Livingston, F. W. Garrett, G. Hirota, M. C. Whitton, E. D. Pisano, and H. Fuchs, "Technologies for augmented reality systems: Realizing ultrasound-guided needle biopsies" [Online]. Available: <http://www.cs.unc.edu/fuchs/publications/TechnoForAug-RealSys96.pdf>
- [8] T. Sielhorst, T. Obst, R. Burgkart, R. Riener, and N. Navab, "An augmented reality delivery simulator for medical training", AMI-ARCS, 2004. [Online]. Available: <http://ami2004.loria.fr/PAPERS/26obetoebiel.pdf>
- [9] A. State, D. T. Chen, C. Tector, A. Brandt, H. Chen, R. Ohbuchi, M. Bajura, and H. Fuchs, "Case study: Observing a volume rendered fetus within a pregnant patient", Tech. Rep. TR94-034, JanuaryAugust, 1994.
- [10] M. Kähäri and D. Murphy, "MARA: Mobile Augmented Reality Applications" [Online]. Available: <http://research.nokia.com/research/projects/mara/index.html>
- [11] "BMW Augmented Reality" [Online]. Available: http://www.bmw.com/com/en/owners/service/augmented_reality_introduction_1.html
- [12] "Augmented Reality Application in Futurescope", 2008. [Online]. Available: <http://www.futuroscope.com/attraction-animaux-du-futur.php>
- [13] Sony, "Eye of judgement" [Online]. Available: http://www.us.playstation.com/PS3/Games/THE_EYE_OF_JUDGMENT
- [14] H. Kato and M. Billinghurst, "Artoolkit library" [Online]. Available: <http://www.hitl.washington.edu/artoolkit/>
- [15] M. Fiala, "Artag, a fiducial marker system using digital techniques," in Computer Vision and Pattern Recognition, 2005. IEEE Computer Society Conference on, vol. 2, 2005, pp. 590-596 vol. 2. Available: <http://www.artag.net/>
- [16] Texas Instruments, "Omap 35xx applications processors", 2008. [Online]. Available: <http://focus.ti.com/lit/ml/sprt457/sprt457.pdf>
- [17] I. Poupyrev, D. S. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani, "Developing a generic augmented-reality interface", Computer, vol. 35, no. 3, pp. 44-50, March 2002

- [18] Kronos Group, "OpenGL. Open Source 2D/3D Interactive Library", 2008. [Online]. Available: <http://www.opengl.org/>
- [19] Intel, "OpenCV. Open Source Computer Vision Library", 2008. [Online]. Available: <http://www.intel.com/technology/computing/opencv/>
- [20] Microsoft, "DirectShow Library", 2008. [Online]. Available: [http://msdn.microsoft.com/en-us/library/ms783323\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms783323(VS.85).aspx)
- [21] Kronos Group, "OpenGL SDK", 2008. [Online]. Available: <http://www.opengl.org/sdk/>
- [22] Intel, "OpenCV library wiki," 2008. [Online]. Available: <http://opencvlibrary.sourceforge.net/>
- [23] Yahoo! Groups, "Open source computer vision library community". [Online]. Available: <http://tech.groups.yahoo.com/group/OpenCV/>
- [24] Microsoft, "MSDN. Microsoft Developer Network", 2008. [Online]. Available: <http://msdn.microsoft.com/>
- [25] Intel, "Core 2 Duo", 2007. [Online]. Available: <http://www.intel.com/products/processor/core2duo/>
- [26] NVIDIA, "Geforce 8400", 2008. [Online]. Available: http://www.nvidia.com/object/geforce_8400_tech_specs.html
- [27] Logitech, "Logitech QuickCam Pro 4000", August 2005. [Online]. Available: <http://web.archive.org/web/20050906014001/www.logitech.com/index.cfm/products/details/US/EN,crid=2204,contentid=5042,detail=2>
- [28] T. Reicher, A. Macwilliams, and B. Bruegge, "Towards a system of patterns for augmented reality systems," 2003. [Online]. Available: <http://citeseer.ist.psu.edu/reicher03towards.html>
- [29] Microsoft, "Visual Studio C++", 2008. [Online]. Available: <http://msdn.microsoft.com/en-us/visualc/>
- [30] Kronos Group, "GL Utility Toolkit", 2008. [Online]. Available: <http://www.opengl.org/resources/libraries/glut/>
- [31] Microsoft, "Component Object Model-based Technologies", 2008. [Online]. Available: <http://www.microsoft.com/com/>
- [32] Microsoft, "Sample grabber filter," 2008. [Online]. Available: [http://msdn.microsoft.com/en-us/library/ms787594\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms787594(VS.85).aspx)
- [33] X. Zhang, S. Fronz, and N. Navab, "Visual marker detection and decoding in ar systems: a comparative study", ISMAR 2002. Proceedings. International Symposium on, 2002, pp. 97-106. [Online]. Available: <http://dx.doi.org/10.1109/ISMAR.2002.1115078>
- [34] S. M. Gerhard, "Robust 2d tracking for real-time augmented reality". [Online]. Available: <http://citeseer.ist.psu.edu/675483.html>
- [35] J. Vitrià, "Introducció a la morfologia matemàtica", CVC Tech Rep, no. 6, January 1996.
- [36] Intel, "OpenCV course examples", CVPR, 2001. [Online]. Available: http://sourceforge.net/project/showfiles.php?group_id=22870&package_id=36092
- [37] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing", RCA Engineer, vol. 29, no. 6, 1984. [Online]. Available: <http://citeseer.ist.psu.edu/adelson84pyramid.html>

- [38] J. Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679-698, November 1986. [Online]. Available: <http://portal.acm.org/citation.cfm?id=11275>
- [39] B. P. Serra and T. Ebrahimi, "Robust feature point extraction and tracking for augmented reality", September 2005.
- [40] C. Harris and M. Stephens, "A combined corner and edge detector", *ALVEY Vision Conference*, pp. 147-151, 1988. [Online]. Available: <http://citeseer.comp.nus.edu.sg/context/35939/0>
- [41] S. J. D. Prince, K. Xu, and A. D. Cheok, "Augmented reality camera tracking with homographies", *IEEE Comput. Graph. Appl.*, vol. 22, no. 6, pp. 39-45, November 2002. [Online]. Available: <http://portal.acm.org/citation.cfm?id=616079.618941>
- [42] B. Boufama and A. Habed, "Registration and tracking in the context of augmented reality", *ICGST-GVIP*, vol. 5, no. 3, March 2005. [Online]. Available: <http://www.icgst.com/gvip/Volume5/Issue3/P1150448002.pdf>
- [43] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations", *International Conference on Computer Vision*, vol. 1, 1999, pp. 666-673.
- [44] V. Vezhnevets and A. Velizhev, *GML Camera Calibration Toolbox*, July 2007. [Online]. Available: <http://research.graphicon.ru/calibration/gml-c-camera-calibration-toolbox-5.html>
- [45] D. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code", in *European Conference on Computer Vision*, 1992, pp. 335-343. [Online]. Available: <http://citeseer.ist.psu.edu/dementhon95modelbased.html>
- [46] Z. Zhang, "A flexible new technique for camera calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, 2000. [Online]. Available: <http://research.microsoft.com/~zhang/Papers/TR98-71.pdf>
- [47] English Wikipedia, "Rodrigues' rotation formula", April 2008. [Online]. Available: http://en.wikipedia.org/wiki/Rodrigues'_rotation_formula

Resum

La Realitat Augmentada és un camp en ple auge investigador. En aquest projecte proposem un entorn amb el qual poder prototipar tant aplicacions d'usuari com algoritmes associats a aquesta tecnologia. En aquesta memòria es recullen l'estudi previ, el disseny i els detalls d'implementació de l'entorn proposat així com una solució específica de Realitat Augmentada associada a aquest entorn basada en visió per computador. Finalment, es presenten els resultats d'una anàlisi de rendiment i de disseny del projecte.

Resumen

La Realidad Aumentada es un campo en pleno auge investigador. En este proyecto proponemos un entorno con el cual poder prototipar tanto aplicaciones de usuario como algoritmos asociados a esta tecnología. En esta memoria se recogen el estudio previo, el diseño y los detalles de implementación del entorno propuesto así como una solución específica de Realidad Aumentada asociada a este entorno basada en visión por computador. Finalmente, se presentan los resultados de un análisis de rendimiento y de diseño del proyecto.

Abstract

Augmented Reality it's a booming research field. In this project we propose an environment with which both prototype user applications and algorithms associated with this technology. This report reflects the previous study, design and implementation details of the proposed environment as well as an Augmented Reality solution associated with this environment based on computer vision. Finally, we present the results of performance and project design analysis.