Visió artificial

Eloi Maduell i García

PID_00184738

Universitat Oberta de Catalunya

Índex

Introducció

La visió artificial o visió per computador són la ciència i la tecnologia que **permeten les "màquines" de veure-hi**, extreure informació de les imatges digitals, resoldre alguna tasca o entendre l'escena que estan visionant.

Les aplicacions de la visió artificial actualment estan molt esteses i van des del camp de la indústria (comptar ampolles, comprovar defectes en una cadena de muntatge, interpretar un TAC mèdic...) i el camp de la medicina (comptatge i cerca de cèl·lules), fins als sistemes més complexos que permeten als robots orientar-se en un entorn desconegut, passant pel reconeixement de patrons de la realitat augmentada, entre moltes altres aplicacions.

Actualment s'utilitzen cada cop més les tècniques de visió artificial en el camp del disseny interactiu mitjançant la interacció amb superfícies (*multitouch*), interacció amb tangibles (objectes) i reconeixement de gestos corporals.

Tots aquests exemples incorporen tècniques de visió artificial.

D'alguna manera, en el context en què estem, aquest **conjunt de tècniques** ens permet **dissenyar interaccions**, de manera que l'usuari utilitza el seu moviment o la seva manipulació d'objectes per a interactuar amb l'aplicació.

Objectius

1. Analitzar les distintes eines, processos i projectes que tenen a veure amb la disciplina de la visió per computadora, aplicada al context de la interacció.

2. Entendre la metodologia d'un projecte de visió per computadora perquè, mitjançant l'aplicació d'aquests conceptes, pugueu dissenyar les vostres pròpies interaccions, basades en visió artificial.

1. Conceptes teòrics

1.1. Llum visible, llum invisible

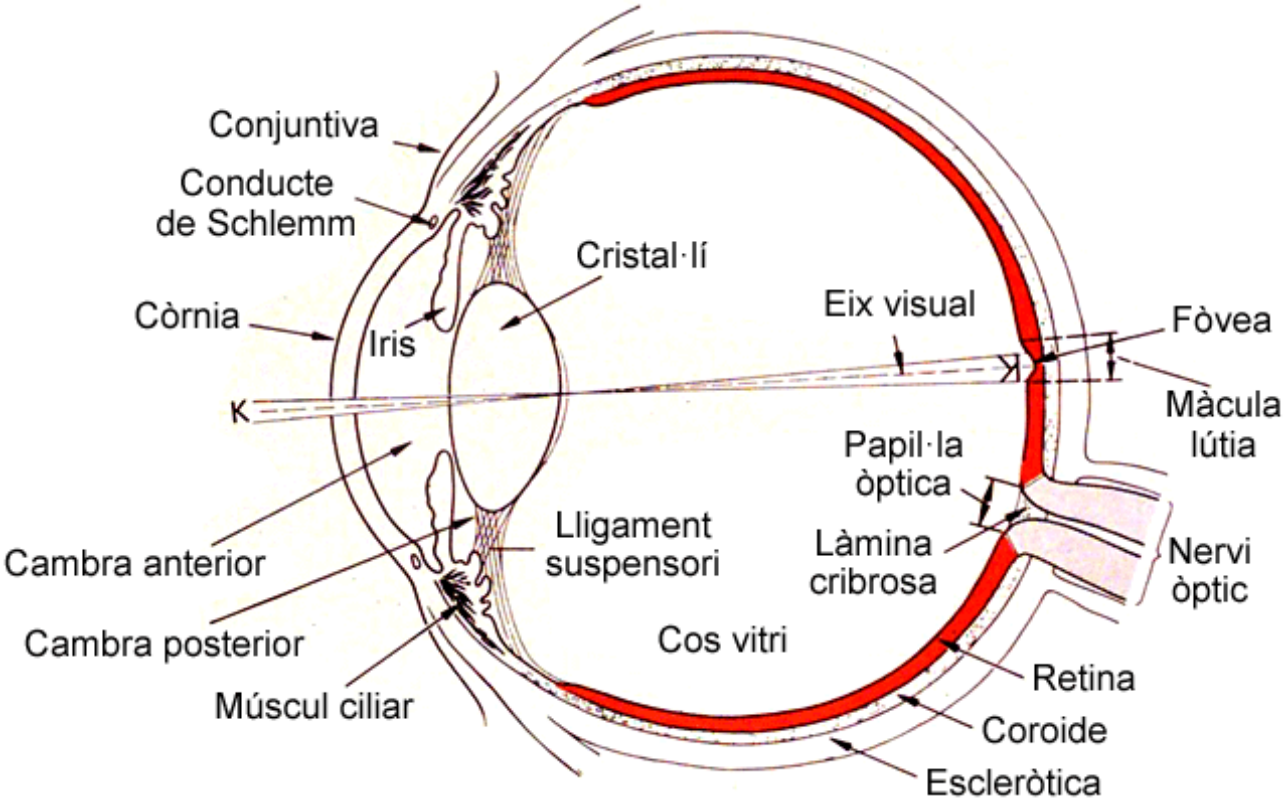
L'ull humà veu una part de l'espectre de tota la llum que il·lumina l'univers. El rang de llum que podem veure l'anomenarem "llum visible".

Això vol dir que hi ha freqüències de llum que no podem veure, però que existeixen, com per exemple els "infrarojos" i els "ultraviolats".

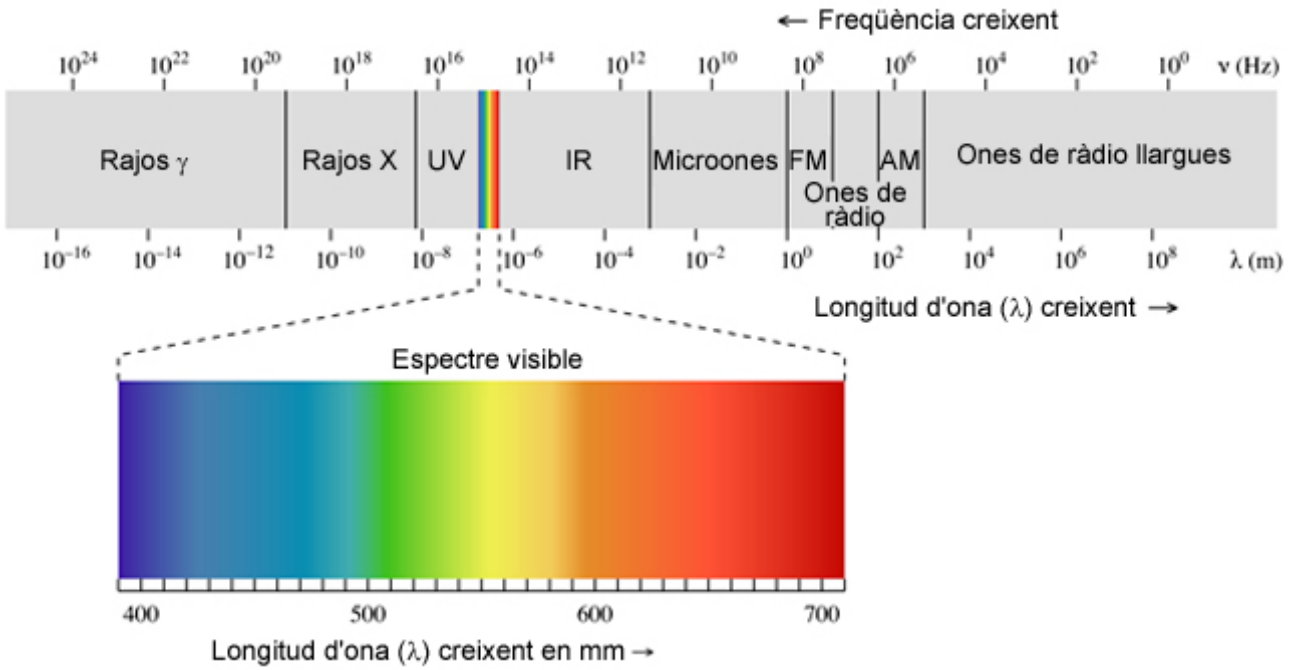
- Els **infrarojos** són els "colors" no visibles a l'ull humà que estan per sota del vermell a nivell freqüencial.
- Els **ultraviolats** són els "colors" no visibles a l'ull humà que estan per sobre del violeta.

Vegeu també

Les imatges impreses en aquest document es poden consultar en color a l'aula virtual de l'assignatura en HMTL o PDF.



Esquema de la morfologia d'un ull humà.
Font: Wikipedia (cc).



Espectrograma de la llum visible i no visible.
Font: Wikipedia (http://en.Wikipedia.org/wiki/Electromagnetic_radiation).

Color	Longitud d'ona
Ultraviolat	< 380 nm
Violeta	380-450 nm
Blau	450-495 nm
Verd	495-570 nm
Groc	570-590 nm
Taronja	590-620 nm
Vermell	620-750 nm
Infraroig	> 750 nm

Abreviatura: **nm** = nanometre (unitat de mesura del sistema mètric, igual a una bilionèsima part d'un metre).

El fet és que la majoria de sensors de càmeres digitals són sensibles a la "llum visible", però també són sensibles (en diferent mesura) a la llum infraroja i/o a la ultraviolada. Ara bé, la llum infraroja no ens és útil per a construir una imatge digital, ja que ens dóna informació d'una freqüència que no podem veure i que, per tant, no té una representació possible en un color.

Per aquesta raó la majoria de càmeres digitals enfocades a poder fer fotografies o fotogrames porten un filtre "anti-IR", o sigui antiinfrarojos, per tallar totes les freqüències per sota de l'espectre visible que ens resultarien en soroll innecessari, i només deixen passar el rang de "llum visible" que volem plasmar amb la càmera.

1.2. La imatge en moviment

Els processos de visió artificial són possibles gràcies a tecnologies basades en la captura de la imatge (càmeres de vídeo, càmeres web) sumades a la capacitat de processament dels ordinadors actuals.

En realitat, la tecnologia de captació de la imatge es comença a gestar al segle XIX, amb la invenció dels **daguerreotips** (o fins i tot abans, amb el descobriment del principi de la **càmera obscura**) i la posterior invenció de la **fotografia**. Descobriments posteriors, com la **cronofotografia** i d'altres precursors del cinema, acaben desembocant en la invenció de les tecnologies de captació de la imatge en moviment:

> tot un seguit de fotografies disparades a una gran freqüència que, reproduïdes després a aquesta mateixa velocitat, provoquen en els espectadors la il·lusió del moviment.

Actualment, la majoria de tecnologies de captació d'imatge en moviment (videocàmeres) funcionen mitjançant l'us de sensors electrònics (CCD i CMOS). Durant la primera dècada del segle XXI s'ha estandarditzat l'us de la fotografia i el vídeo digitals, la qual cosa permet a un relatiu baix cost la gravació d'imatges en alta resolució i amb un elevat nombre d'imatges per segon (fps).

1.3. La imatge digital

1.3.1. Matrius de píxels

En el món digital, les imatges es representen com una matriu bidimensional de píxels, en què cada píxel pot adquirir **valors de color codificats** amb tres paràmetres (**R**: *red*, **G**: *green*, **B**: *blue*).

Tot i que s'ha heretat del món de la imatge analògica, normalment treballarem amb imatges de proporció 4 × 3 (4 unitats d'amplada per 3 unitats d'alçada); és el cas de resolucions estàndard com 640 × 480 px, 800 × 600 px, 1.024 × 768 px.

16 × 9

En alguns casos podem trobar imatges en el format 16 × 9, ja que avui dia s'han popularitzat les videocàmeres HD en format panoràmic.

Freqüentment utilitzarem imatges a baixa resolució (entre 160 × 120 i 640 × 480 píxels) com a font d'anàlisi dels processos de visió artificial, ja que en la majoria de casos no necessitem excessiu detall en les imatges per tal d'efectuar-ne una anàlisi orientada a la visió per computador.

1.3.2. Bytes, bits i colors

1 píxel normalment s'expressa mitjançant tres nombres enters (R, G, B), que representen les components vermella, verda i blava de tot color. Aquests valors de R, G i B normalment s'expressen en un rang de 8 bits o sigui de valors entre 0 i 255.

Exemple

Per exemple, un píxel que tingui uns valors RGB de (255, 0, 0) ens indica un color vermell pur.

Un píxel que tingui uns valors RGB (255, 0, 255) ens indica un color produït per una mescla del vermell pur i el blau pur; en aquest cas, per tant, obtindrem un color lila intens.

1.3.3. *Frame rate*

El *frame rate* fa referència al nombre d'imatges per segon. És la mesura de la freqüència a la qual un reproductor d'imatges mostra diferents fotogrames (*frames*).

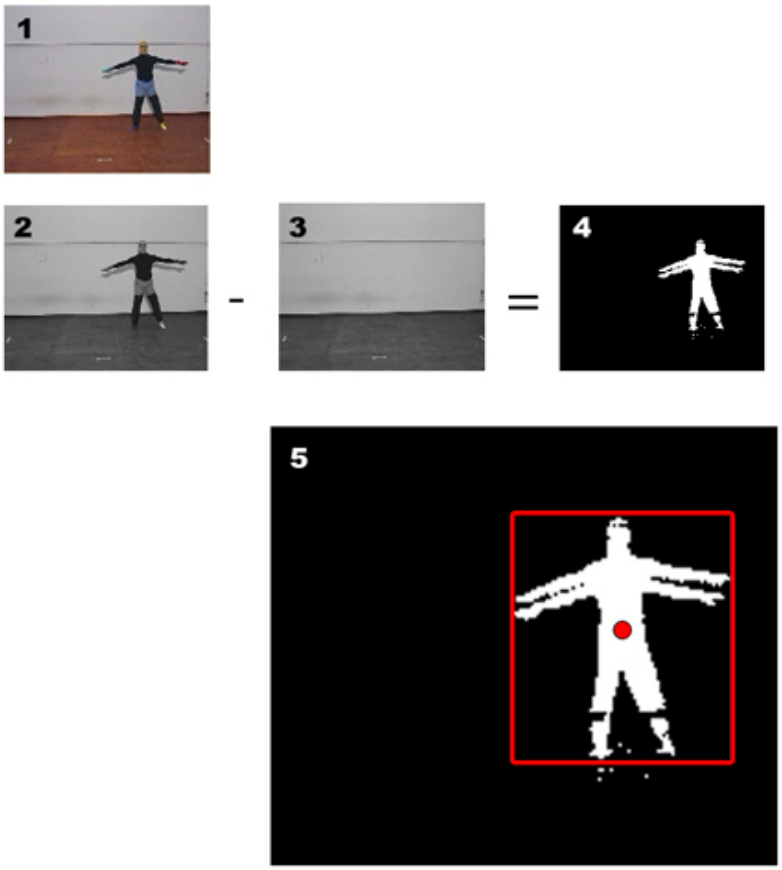
En informàtica aquests fotogrames estan constituïts per un nombre determinat de píxels que es distribueixen al llarg d'una xarxa de textures. La freqüència dels fotogrames és **proporcional** al nombre de píxels que s'han de generar, i que incideixen en el rendiment de l'ordinador que els reprodueix.

La freqüència d'actualització de les imatges oscil·la, en el entorn digital, entre els 15-60 fps (*frames* per segon). El rang 25-30 fps és el més comú.

Segons les nostres necessitats, haurem de triar tecnologies de captació de la imatge amb un *frame rate* específic. Si volem analitzar i extreure dades d'objectes o d'usuaris que es mouen a grans velocitats (cotxes, ocells, corredors de futbol...), segurament necessitarem un sistema de captació de vídeo amb més resolució temporal (un *frame rate* més alt: més fotogrames per segon).

1.4. Un exemple bàsic

Tot i que entrarem en profunditat en la matèria en els apartats següents, és convenient fer una breu anàlisi d'un típic procés de visió artificial per tal que, d'ara en endavant, pugueu analitzar amb una certa perspectiva els conceptes que anirem introduint. Fixeu-vos en aquest exemple esquema:



1) Imatge original
2) Imatge processada en escala de grisos
3) Imatge neta del fons
4) Imatge resultant de la sostracció del fons i la binarització
5) Extracció de dades (àrea del *blob* i centre de coordenades)
Font: www.eyesweb.org

En aquest exemple podeu observar alguns dels processos típics de la visió artificial.

Sovint necessitem aconseguir una imatge molt clara i simple de l'objecte o usuari del qual volem efectuar un seguiment. Per tal de no carregar el processador dels ordinadors, és habitual aplicar els algorismes d'anàlisi sobre imatges de color binari (1 bit: blanc i negre), de les quals es pot extreure el centre de masses de l'àrea de píxels blancs, els contorns...

Per tal d'aconseguir aquesta imatge simple i clara, efectuem processos diferents, com podeu observar en aquest exemple: la sostracció de fons o la binarització, entre altres.

Un cop aconseguida aquesta imatge binària final, els algorismes de visió ens permeten extreure un seguit de dades, com veurem a continuació. En aquest cas particular, s'extreuen les coordenades del centre de la massa blanca de píxels i l'àrea que l'envolta (punt i línies vermelles).

2. L'espai i les eines

2.1. La càmera

Per a començar a treballar amb la visió artificial, necessitem un **dispositiu sensible a la llum visible** que ens permeti d'emmagatzemar les imatges en format digital. En altres paraules: una càmera web, càmera de vídeo o capturadora analògica.

Exemple de càmera de visió artificial del sector industrial fabricada per Point-Grey
Font: http://doc.instantreality.org/

Les càmeres web i la majoria de càmeres de vídeo es poden connectar directament als ordinadors a través dels ports USB, FireWire o Thunderbolt, i en podem capturar fotogrames en **temps real**. En tot cas, també podem utilitzar una capturadora analògica que, a partir d'un senyal de vídeo analògic, ens permetrà de capturar imatges i processar-les.

Exemple de càmera web fabricada per Logitech
Font: http://doc.instantreality.org/

Un cop ja tenim el dispositiu de captura en funcionament, hem de considerar tota una sèrie d'aspectes referents a l'entorn on es farà la interacció, ja que la **variació en la il·luminació**, la complexitat de la imatge o altres factors poden dificultar molt el procés de visió artificial.

Penseu que el procés d'"ensenyar" a un ordinador a prendre decisions per mitjà de la "visió" implica moltes dificultats i, per tant, és molt important treballar en entorns on la llum i l'escenari que es volen analitzar siguin com més senzills i estables millor.

2.2. Il·luminació de l'escena

L'adquisició d'imatges per part d'una càmera varia molt segons la il·luminació de l'escena. Un canvi prou fort en l'ambient lumínic pot fer que tot el sistema de visió artificial funcioni de manera molt diferent.

Per tant, sempre que sigui possible treballarem en entorns on l'ambient lumínic sigui com **més estable** millor.

En casos en què no es pot aconseguir un ambient estable (com ara una aplicació a l'aire lliure), haurem d'anar adaptant d'alguna manera el sistema d'adquisició d'imatges a les condicions canviants, mitjançant **algoritmes adaptatius** que permetin analitzar periòdicament els canvis d'il·luminació de l'entorn i adaptar el sistema de visió a les noves condicions.

Algunes càmeres com la que porta el comandament de la Wii o moltes del sector industrial són especialment sensibles als infrarojos, ja que ens permeten treballar en un entorn sense llum visible, en l'obscuritat, sempre que tinguem alguna font de llum infraroja.

Exemple

Per exemple, molts sistemes de videovigilància incorporen, a més de videocàmeres sensibles a la llum IR, un anell de LEDS que emeten llum infraroja, invisible a l'ull però visible per a la càmera, la qual cosa permet la visibilitat en entorns d'obscuritat aparent.

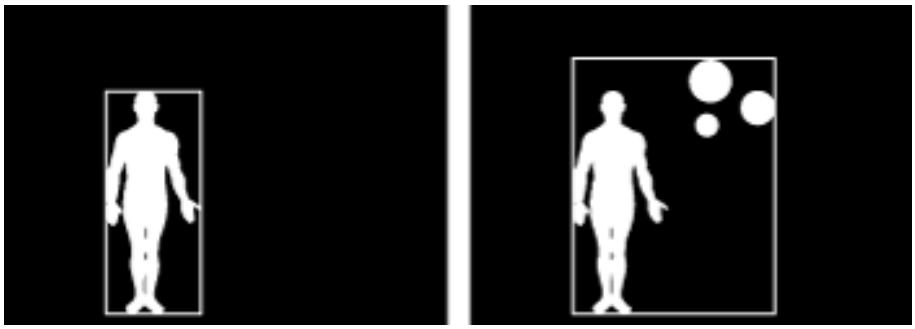
Aquesta particularitat s'aprofita en molts casos en el disseny d'interfícies interactives.

Per exemple, posem el cas d'un teatre on un actor a les fosques descriu una trajectòria per l'escenari i volem projectar al terra el seu recorregut. En principi, amb una càmera normal, com que estem a les fosques, no podríem captar-hi res. Per a poder treballar en condicions de foscor, el que faríem és il·luminar l'escena amb llum infraroja, que no és visible, però que sí que és visible per a la càmera sensible als IR (per exemple una càmera web a la qual hem tret el filtre anti-IR esmentat anteriorment). Així doncs, una càmera d'aquestes característiques en una situació com aquesta obté una imatge molt neta del cos de l'actor (en blanc) sobre el fons (negre) de l'escenari, una imatge perfecta per a analitzar amb visió artificial, ja que el cos no interfereix amb el fons.

La llum infraroja es pot generar de diverses maneres: hi ha focus de LED infrarojos que ens permeten "bombardejar" un espai amb llum IR. Una altra manera menys costosa és utilitzar una llum de teatre incandescent, a la qual posarem tres filtres: un de vermell, un de verd i un de blau (color purs). L'efecte de posar un filtre vermell fa que la llum surti tintada de vermell, és a dir, que eliminem tots els colors excepte el vermell de la llum filtrada. Si sobre aquest filtre hi posem un filtre blau, estarem eliminant tots els colors (incloent-hi el vermell que ens havia quedat) excepte el blau (que ja no estava a la llum filtrada perquè l'havia eliminat el filtre vermell). I sobre aquests dos filtres hi afegim el filtre verd, amb resultats semblants. Per tant, hem eliminat tota la llum visible amb els tres filtres, però no hem eliminat les freqüències infraroges amb els filtres. El que hem fet és transformar una llum incandescent en una llum infraroja.

2.3. Projeccions de vídeo i visibilitat

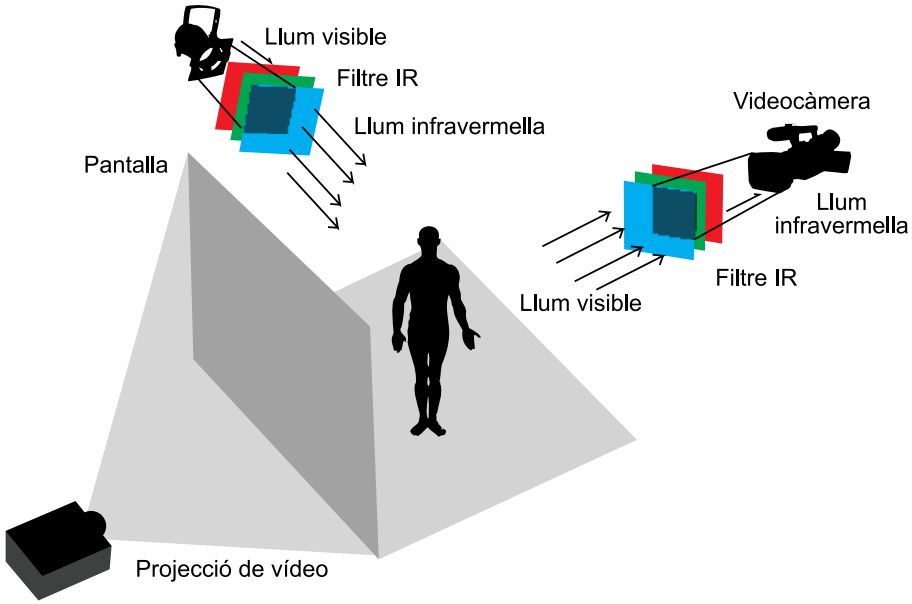
En un entorn interactiu ens podem trobar sovint amb la necessitat de tenir en un mateix espai un sistema de visió artificial i una projecció de vídeo. Com veurem una mica més endavant, la complexitat dels algorismes de visió artificial podria fer que un sistema hagi de discernir entre una projecció de vídeo interactiva i l'usuari que la manipula, cosa molt dificultosa si totes dues "imatges" estan integrades en un mateix espai. Com en l'apartat anterior, gràcies a comprendre la resposta de la llum i les seves propietats, podem solucionar aquestes dificultats.



Si no bloquegem l'entrada de llum visible a la càmera, trobarem problemes en el context d'una interacció amb videoprojeccions, ja que el sistema CV reconeixeria els elements projectats (cercles blancs) com a part integrant dels *blobs* analitzats.

La llum projectada per un projector evidentment és visible a l'ull humà i/o per les càmeres digitals normals. En el cas que plantegem, volem obtenir una imatge de l'usuari que interactua amb la pantalla de videoprojeccions, però no dels continguts de la videoprojecció. Per tal d'assolir aquesta fita, podem fer ús de la tècnica descrita en l'apartat anterior, referent a la llum IR, fent servir una càmera que "vegi" la llum de l'espectre infraroig.

Si a aquesta càmera hi afegim un filtre que bloquegi la llum visible, el que aconseguirem és que la càmera no sigui sensible a la llum visible i només capti la llum IR. Per tant, aquesta càmera podrà veure l'usuari manipulant el sistema, ja que el seu cos fa rebotar la llum IR, però no podrà veure la projecció (que es troba només dins el rang de llum visible).



Esquema de la configuració escènica comentada, llum visible filtrada a IR, càmera de vídeo filtrada a IR i retroprojector de vídeo sobre la pantalla de fons.

Reflexió

Com a corol·lari d'aquest exemple, podem concloure que un projector de vídeo no genera llum infraroja, ja que els projectors porten un filtre de llum anti-IR que no permet que els projectors "projectin" llum en l'espectre IR.

2.4. OpenCV

OpenCV o Open Computer Vision és un conjunt de biblioteques de programació que ens permeten dur a terme tot el procés que hem vist anteriorment dins d'un ordinador: des de l'adquisició de la imatge fins a l'extracció de la informació.

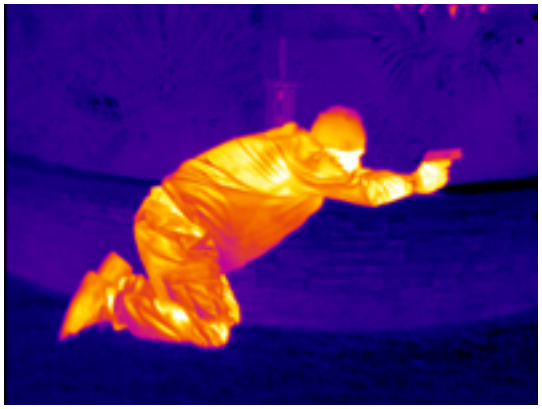
La gran majoria d'aplicacions dins el programari lliure relacionades amb la visió artificial estan basades en OpenCV. Aquestes biblioteques, les va desenvolupar Intel com a extensió de les biblioteques IPL, des del 1999 com a projecte de codi obert i l'any 2007 s'arribà a publicar la versió 1.0. Aquestes biblioteques són la base d'algorismes i funcions que finalment podrem aplicar en les nostres aplicacions.

2.5. Entorns de programació

Per a desenvolupar una aplicació que treballi amb visió artificial hem de treballar en un entorn de programació que ens permeti integrar les biblioteques d'OpenCV i combinar-les amb la resta del sistema interactiu. Dins d'aquest sac hi podríem posar entorns com Processing, openFrameworks, MAX/MSP,

PureData... Tots aquests sistemes de programació ens permeten executar algoritmes i funcions de visió artificial i a la vegada generar un sistema interactiu que treballi amb gràfics, música, vídeo, etc.

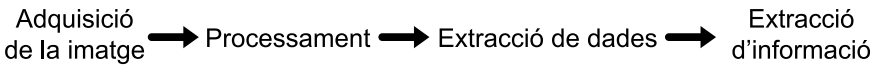
Dins el sector industrial, mèdic i militar més professional hi ha una gran quantitat de solucions i alternatives tant de maquinari (càmeres tèrmiques, càmeres lineals, càmeres d'alta resolució o alta velocitat) com de programari amb solucions alternatives a l'OpenCV.



Exemple de càmera tèrmica. La imatge es construeix segons la temperatura de cada punt.
Font: www.x20.org

3. Dissenyant interaccions: conceptes, algoritmes i funcions. Kinect

Les tècniques de visió artificial segueixen normalment un procés des de l'adquisició de la imatge fins a la presa de decisions segons la informació processada. El resumim a continuació:



3.1. Adquisició de la imatge

Una imatge digital és produïda per sensors digitals presents en càmeres i altres dispositius digitals que generen una imatge bidimensional (2D), és a dir, un conjunt de $N \times M$ **píxels** o valors de color o intensitat d'un cert valor que representen l'**espai** que volem analitzar.

En el món de la interacció **podem utilitzar càmeres digitals de diferents tipus** (càmeres web, DV o USB). En el fons, cada tipus d'aplicació pot necessitar un tipus o un altre de càmera segons les necessitats.

3.2. Processament de la imatge

Abans d'extraure informació directament de la imatge, s'acostuma a fer un processament previ de la imatge per tal d'aconseguir una altra imatge que ens permeti de fer el procés d'extracció de dades d'una manera més senzilla i eficient.

Exemple

Per exemple, el processament pot incloure funcions per a modificar la lluminositat i el contrast, reescalar la imatge, nivells de color, corbes, binarització, desenfocament (*blur*), etc.

Com a exemple, el fet d'aplicar un desenfocament en una imatge ens permet d'alguna manera "arrodonir" els contorns de les formes que hi apareixen i això ens pot ser útil en diverses ocasions. Modificar la lluminositat per a tenir una imatge més contrastada ens pot ajudar a donar més importància visual a la part que volem analitzar. Reescalar i reenquadrar la imatge ens permet de centrar el procés de visió artificial només sobre la porció de la imatge que ens interessa realment, i així ens estalviem un processament innecessari.

A continuació analitzarem alguns algoritmes de processament de la imatge usats freqüentment en la visió artificial.

Aquesta serà la imatge d'exemple original que utilitzarem per a explicar diferents funcions del processament de la imatge. Té una resolució de 800 píxels horitzontals per 800 de verticals.

Imatge original
Font: http://opencv.willowgarage.com

3.2.1. Escala de grisos

En molts casos el color de la imatge no ens aportarà cap informació rellevant per a interactuar; per tant, en casos així s'acostuma a descartar la informació de color de la imatge i es transforma en **imatge de tons de grisos**.

D'aquesta manera "estalviem" molta informació i els càlculs seran molt més senzills de computar. Per exemple, si estem analitzant el moviment dins d'una imatge de color, ens podem estalviar aproximadament 2/3 parts dels píxels que s'han de tractar si en descartem els components de color i treballem simplement amb la imatge d'intensitat lluminosa o nivells de grisos.

Transformem la imatge de proba de color (R, G, B) a tonalitat de grisos. És a dir, només amb una dimensió de color (L) de luminància.

Per exemple, la imatge original ocupava en la memòria, en nombre de bytes:

800 píxels × 800 píxels × 3 bytes (R, G, B) = **1.920.000 bytes**.

En canvi, la imatge transformada en blanc i negre ocuparà en nombre de bytes:

800 píxels × 800 píxels × 1 byte (L) = **640.000 bytes**.

3.2.2. Binarització mitjançant llindar

El procés de binarització per llindar parteix d'una imatge en tons de grisos i a partir d'un valor definible de llindar d'intensitat de llum anomenat *threshold*, la imatge es transforma en una imatge binària, és a dir amb píxels blancs o píxels negres.

Si el píxel analitzat tenia un valor d'intensitat inferior al llindar, el píxel quedarà en negre i si la intensitat era més elevada que el llindar, el píxel quedarà en blanc. D'alguna manera, això ens permet descartar els tons mitjans grisos i facilita molt la computació de certs algorismes, ja que una altra vegada hem transformat la imatge en quelcom molt més lleuger i fàcil de processar.

Transformem la imatge de prova de color (R, G, B) a blanc i negre; és a dir, només amb dos colors: blanc absolut i negre absolut.

Tal com hem vist, la imatge original pesava 1.920.000 bytes (1 byte = 8 bits); per tant, pesava 15.360.000 bits.

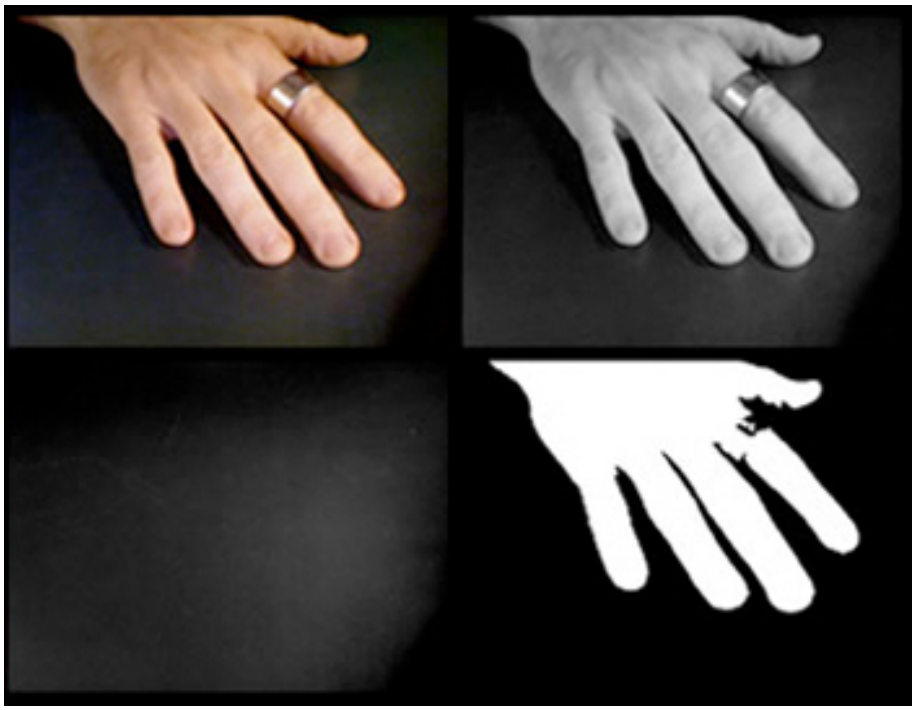
En canvi, la imatge transformada amb llindar de blanc i negre ocuparà en nombre de bits:

800 píxels × 800 píxels × 1 bit = 640.000 bits, és a dir, 24 vegades menys en nombre de bits que l'original.

Velocitat i resolució de la imatge

Molts dels processos de visió artificial es basen a treballar per cada píxel de la imatge internament; per tant, si amb la binarització aconseguim 24 vegades menys de bits a

processar, això ens farà córrer els algoritmes molt més ràpid. Així doncs, un factor determinant en la velocitat dels càlculs que es generen en les aplicacions de visió artificial és la resolució de la imatge en píxels. Una imatge de 320 × 240 píxels serà processada molt més ràpidament que una imatge a 1.024 × 768 píxels.



Font: www.openframeworks.cc.

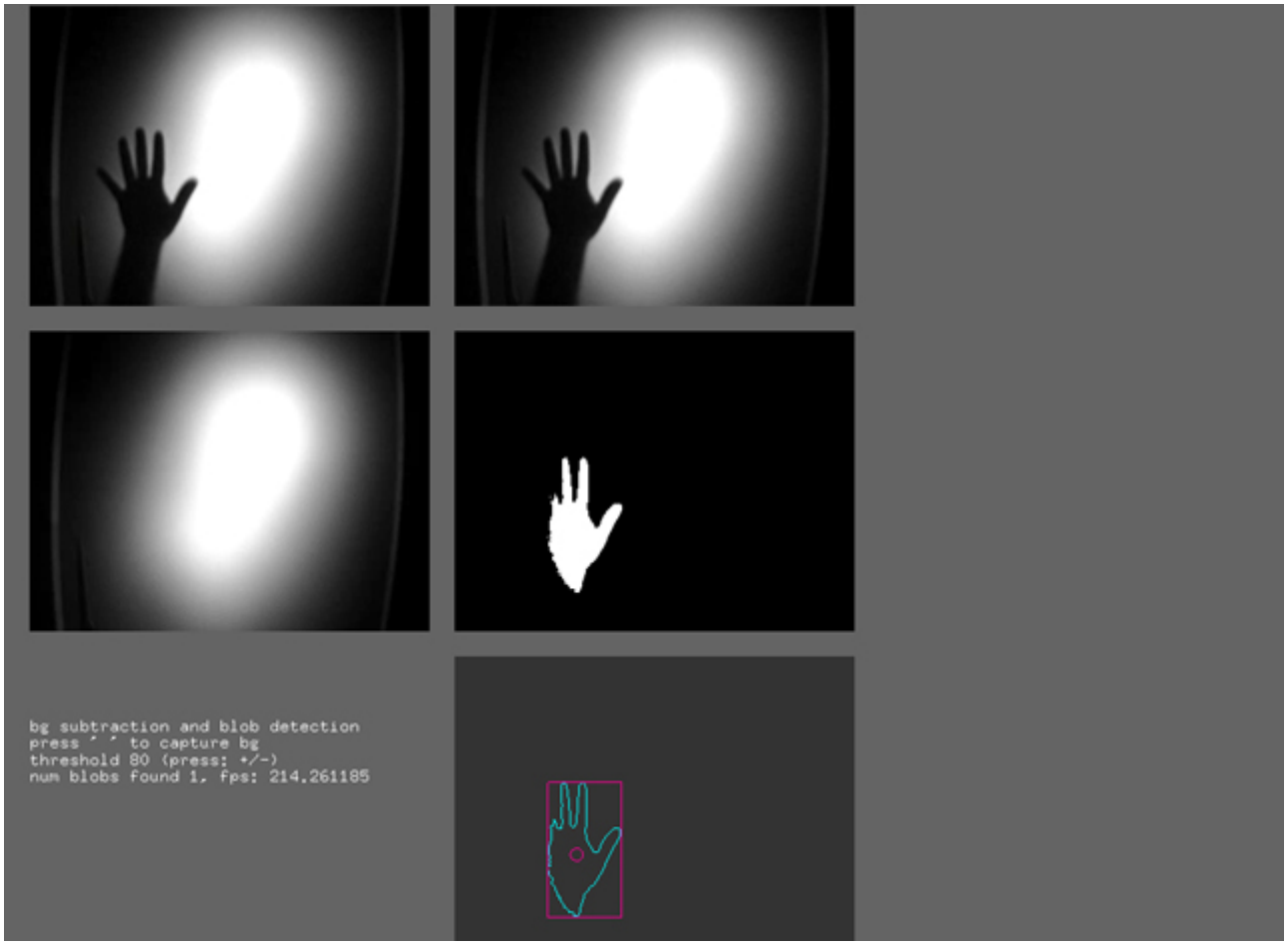
En aquesta imatge podem veure les tres representacions fins ara esmentades. A dalt a l'esquerra, tenim la representació en color del fotograma capturat per la càmera; a dalt a la dreta, la transformació a blanc i negre de la mateixa imatge, i a baix a la dreta, la representació binària de la imatge en blanc i negre. El llindar en aquesta binarització s'ha fixat en un valor que permet distingir fàcilment la mà (blanca) del fons (negre).

3.2.3. Sostracció del fons *o background substraction*

La sostracció del fons o *background substraction* és una tècnica de processament de la imatge que permet "restar" una imatge del fons de l'escena amb el fotograma actual, i obtenir, doncs, el fotograma actual "menys" el fons. D'aquesta manera podem aïllar els objectes que han variat respecte de la imatge del fons d'una escena i determinar si hi ha hagut moviment.

Reflexió

Cal tenir en compte que aquesta tècnica requereix poder "capturar" la imatge de "fons". La seva utilitat és en entorns relativament controlats, on podem conèixer com serà el fons durant tot el processament, o bé que ens hi podem anar adaptant.



Font: www.openframeworks.cc.

En aquesta imatge veiem tot el procés. La imatge de l'esquerra a dalt és el que "veu" la càmera, i la de l'esquerra a baix és la imatge del "fons" o *background* que s'ha de restar. Un cop feta la resta i binaritzada, veiem el que es veu a la imatge del mig a la dreta, on pràcticament només apareix la mà en una imatge binària (només amb píxels 100% blancs i 100% negres). Aquesta darrera imatge binaritzada i restada del fons ja ens deixa casi la silueta perfecta de la mà per aplicar-hi el pas següent, que és identificar la "taca" o *blob* de la mà per tal d'obtenir-ne certa informació, tal com veurem en el l'apartat següent.

3.2.4. Altres operacions morfològiques

En moltes situacions, un cop hem obtingut una imatge binària (en blanc i negre), observem que apareix **soroll** a la imatge, fruit dels canvis en la il·luminació ambient iI els petits reflexos d'aquesta llum en els diferents objectes i cossos a l'escena. Aquest soroll sol tractar-se simplement de píxels que apareixen caòticament en la imatge i que poden destorbar seriosament la capacitat dels algoritmes de visió per a reconèixer *blobs* i patrons gràfics.

Per tal d'eliminar aquests petits i molestos píxels, hi ha un seguit d'algorismes de "neteja" de la imatge que ens ajuden a acomplir la tasca d'obtenció d'una imatge binària el més neta possible.

1) Mediana (*Median*)

L'algorisme de la mediana s'aplica sobre les imatges en escala de grisos i provoca una certa suavització de la imatge, que és útil a l'hora de simplificar contorns i àrees irregulars.

L'algorisme de la mediana divideix la imatge en un conjunt de divisions de radi definible i calcula el valor de lluminositat mitjà de cadascuna de les subdivisions. Un cop calculat, substitueix els píxels que hi ha a dins de cada divisió per una única taca grisa amb el valor de lluminositat mitjà dels píxels originals.

1. Imatge original

2. Aplicació de la mitjana

2) Erosionar/Dilatar (*Erode/Dilate*)

Els algorismes d'erosió i dilatació se solen aplicar en sèrie. Sobre la imatge "sorollosa" s'aplica l'algorisme d'erosió, que contrau els contorns de totes les àrees blanques un nombre determinat de píxels i elimina completament les àrees de píxels blancs més petites i irrellevants.

Un cop aplicat l'algorisme d'erosió, s'hi aplica l'algorisme de dilatació, que ajuda a recuperar la mida original de les àrees importants i expandeix els contorns de les àrees blanques tants píxels com sigui necessari.

1. Imatge sorollosa

2. Imatge erosionada

3. Imatge dilatada

3.3. Extracció de dades

Un cop tenim la imatge preparada, s'hi apliquen una sèrie d'algoritmes per a extreure les dades de la imatge. Podem reconèixer en la imatge per exemple: les línies, els cercles, les taques, les cantonades, el moviment, un determinat color, o bé reconèixer certs patrons d'imatge prèviament determinats.

Un cop reconeguts en la imatge el que volíem extreure (sigui per cada línia, cercle o taca reconeguts) de cada element, en podrem esbrinar la posició, la velocitat de moviment, la mida o el color dominant ...

A continuació, analitzarem un seguit d'algoritmes freqüentment usats per a la extracció de dades en el context de projectes relacionats amb visió artificial:

3.3.1. Detecció de *blobs*

Anomenem *blob* als punts o les regions de píxels que són més lluminosos (o més foscos) que els voltants i que, per tant, formen una "taca" aïllada de la resta de la imatge.

Les eines de ***blob detection*** ens permeten analitzar les "taques" d'una imatge binària (blanca i negra) i extreure informació de cada una, com per exemple: la mida de la taca, el centre de masses de la taca, el seu contorn...

Aquest tipus de funcions són útils en molts casos, per exemple, per a comptar i localitzar quants objectes tenim en una escena. A partir que els hàgim localitzat, en podrem seguir la trajectòria o analitzar-ne el moviment. Per tant, aquesta tècnica s'utilitza en molts processos interactius.

Detecció de *blobs*

Considereu que la detecció de *blobs* requereix normalment d'entrada una imatge ja binaritzada amb les funcions de processament de la imatge que acabem de veure.

3.3.2. *Frame difference*: *tracking* de moviment

Aquesta tècnica ens permet obtenir dades del moviment que hi ha en l'escena que estem analitzant, tant de la quantitat de moviment com de la seva direcció.

La tècnica consisteix bàsicament a restar un fotograma capturat per la càmera amb l'anterior. La resta ens dóna una referència del que ha canviat entre totes dues imatges. Normalment aplicarem aquesta tècnica després d'haver fet la sostracció del fons, és a dir, que haurem eliminat molta informació del fons.

En cas que no hagi canviat res entre un fotograma i el seu anterior, la resta ens donarà una imatge pràcticament negra.

Si per exemple en la càmera estem veient un objecte en moviment, la resta entre dos fotogrames consecutius ens donarà les parts de la imatge que han canviat. Si la imatge ha canviat molt (perquè l'objecte es desplaçava ràpidament), la resta serà una taca força gran. Si la imatge ha variat molt poc, l'operació de resta ens donarà una taca molt petita.

Per acabar d'extreure'n tota la informació, el que farem és una detecció de *blobs*. Com a resultat, tindrem per exemple la mida de la taca del *blob*, que serà una mesura de la quantitat de moviment. O bé, si comparem els centres de masses de la taca o *blob* en dos fotogrames consecutius, en podrem extreure una mesura de la direcció del moviment.

3.3.3. *Color tracking*

El tracking de color ens permet fer el seguiment d'una àrea de píxels d'un determinat color (és a dir, d'un valor específic RGB).

Tot i que aquesta tècnica no s'inclou en la implementació clàssica de les biblioteques OpenCV, és força popular. De fet, es tracta més aviat de la suma d'un conjunt de tècniques, que impliquen la extracció selectiva de canals de color, per tal d'aïllar-ne el color seleccionat i obtenir una imatge binària en què només el color de què s'ha fet el *tracking* apareix com un *blob* o una taca blanca de píxels.

Generalment, quan fem un *tracking* de color, seleccionem el color que volem resseguir i definim la mida de l'àrea al voltant d'aquest píxel. L'algorisme haurà d'anar reescanejant permanentment aquesta àrea per tal de no perdre el seguiment en cas que el píxel original canviï de color. Com més gran és aquesta àrea de seguretat, més dificultats tindrà l'ordinador per a fer tots els càlculs, ja que es tracta d'una operació amb gran demanda de capacitat de càlcul per part del processador.

Gran popularitat del *tracking*

El *tracking* de color és molt popular també en entorns de programari més enllà de les aplicacions d'interactivitat en temps real. Se'n fa un ús intensiu en molts programaria de tractament professional de vídeo, per tal de resseguir un punt d'una gravació de vídeo, i afegir-hi efectes de postproducció, estabilitzar els moviments involuntaris de la càmera, etc.

3.3.4. Cercador de cares o *face tracking*

Hi ha tot un seguit de tècniques que estan enfocades cap al reconeixement de patrons o formes concretes en una imatge. Aquestes tècniques ens permeten definir quins patrons o imatges buscarem dins l'escena. És d'aquesta manera com funcionen la majoria de tècniques de cercador de cares.

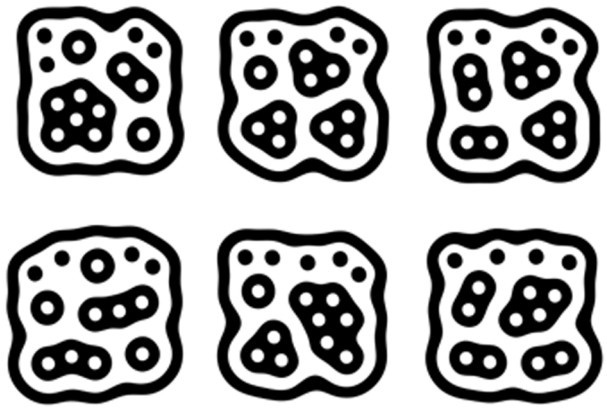
Aquesta funció detecta si hi ha "cares humanes" dins la imatge i ens permet extreure certa informació de la cara, com ara la posició i la mida, i alguns aspectes morfològics (que són la base per a poder identificar les cares de diferents usuaris).

Imatge d'exemple de cercador de cares dins el projecte "AAM Fitting Algorithms"
Font: Carnegie Mellon University - Robotics Institute.

3.3.5. Cercador de característiques o *feature tracking*

A continuació denominarem sota un mateix nom tot un seguit de tècniques basades en l'extracció de *features* o "característiques" d'una imatge, que bàsicament són punts de la imatge fàcilment recognoscibles per certs algoritmes (detecció de cantonades per exemple).

Una altra aplicació d'aquestes tècniques permet localitzar els **fiducials** d'una escena. Els fiducials serien patrons especialment dissenyats per a ser reconeguts, com per exemple els fiducials que utilitza l'instrument interactiu Reactable, desenvolupat per l'equip de Sergi Jordà (MTG-UPF).



Marcadors fiducials del sistema Reactivision, utilitzats en la Reactable
Font: reactivision.sourceforge.net.

3.3.6. Realitat augmentada

Com heu anat veient, anem acumulant unes tècniques a sobre d'unes altres i anem fent accions cada vegada més complexes. De l'anterior concepte de *feature tracking* es desprèn en certa manera la tècnica de la realitat augmentada o *augmented reality* realitat augmentada.

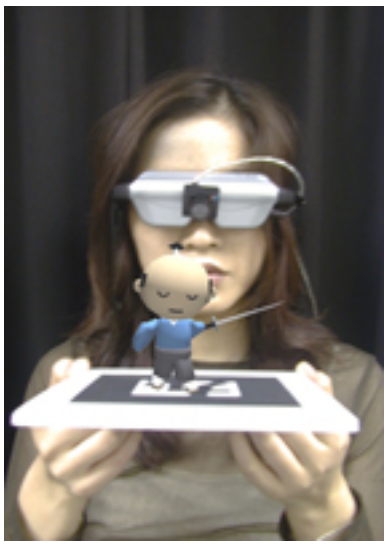
Per realitat augmentada entenem totes les tècniques que d'alguna manera integren una imatge virtual dins d'una imatge d'un entorn real, i evidentment en temps real de manera interactiva.

Una de les biblioteques més utilitzades s'anomena **ARToolKit**, i s'hi pot treballar des de molts entorns de programació creativa.

Exemple de marcador d'un sistema de realitat augmentada

Una de les aplicacions més conegudes és la capacitat d'introduir un element 3D interactiu dins de la imatge capturada per una càmera. Aquesta integració és coherent amb el punt de vista de la càmera i ens fa l'efecte que els objectes 3D són presents en l'entorn vist per la càmera. Evidentment, aquesta tècnica requereix una càmera per a tenir sentit i sempre veurem el món 3D integrat en la visió de la cèmera en un monitor.

Aquesta imatge és un marcador o *marker* per a un sistema de realitat augmentada. Aquest simple patró és analitzat pel sistema de visió artificial i en podem treure la posició i l'orientació relatives a la càmera. A partir d'aquí només ens cal vincular un món 3D virtual en relació a aquesta marca, copiant-ne la perspectiva i l'orientació.



Típica aplicació de la realitat augmentada. Es genera una escena 3D en relació amb la posició del marcador
Font: artoolkit.sourceforge.net

Aquí podem veure una aplicació molt senzilla de realitat augmentada: l'usuari té a les mans un paper amb un marcador imprès, el sistema de visió reconeix el marcador, la seva orientació i genera una figura 3D sobre el marcador, coherent amb la posició, rotació i inclinació del marcador.

3.4. Kinect

Des que va aparèixer al mercat la càmera Kinect de Microsoft es va posar a l'abast la possibilitat de treballar en el camp de la visió volumètrica o tridimensional.

> Aquest tipus de càmeres volumètriques ens permeten d'adquirir una imatge plana en color i una imatge de profunditat, és a dir, ens permet saber a quina distància (o profunditat) de la càmera es troben cada un dels píxels de la imatge.

A més a més, si dins la imatge apareix la figura d'un cos humà, la mateixa càmera és capaç de reconèixer la postura de la figura humana i enviar-nos al nostre programari la posició i l'orientació de tronc, cap, braços i cames.

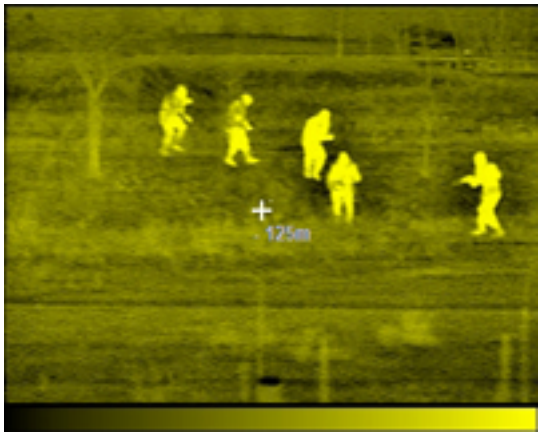
Al cap de pocs dies de la seva aparició al mercat, a Internet van aparèixer persones que havien aconseguit desencriptar el protocol USB de la Kinect fent servir mètodes d'enginyeria inversa i els van publicar a Internet. En poques setmanes una gran quantitat d'entorns de programació interactiva permeti-en interactuar amb la Kinect: Processing, openFrameworks, Quartz Composer, Max/MSP, EyesWeb, Cinder, vvvv, Flash, etc. Així doncs, un producte pensat per al mercat de les consoles XBox es va començar a utilitzar extensament en el món del disseny d'interactius.

Fotografia infraroja en què es mostra el núvol de punts infrarojos que emet la Kinect per a obtenir la profunditat de la imatge.
Font: www.mattcutts.com

El fet que la càmera Kinect sigui un producte comercial massiu permet que el seu preu sigui molt assequible comparat amb les prestacions que ens dóna en un sistema de visió artificial. El fet de poder disposar, d'una banda, d'una imatge amb profunditat i, d'una altra, del reconeixement de figures humanes ens facilita moltíssim el treball d'interacció basat en la visió.

> Per exemple, amb una imatge amb profunditat d'un cos, podem demanar a l'aplicació que només treballi sobre els píxels que hi ha entre 1m50 i 1m30 de distància de la càmera. Si, per exemple, l'usuari és a 1 m de la càmera i mou les mans endavant fins que entrin dins el rang d'1m50 a 1m30, el sistema podrà treballar amb una imatge on només les mans apareixen dibuixades; per tant, les tinc fàcilment aïllades per a poder-les detectar amb un *blob-tracking* i extreure'n la posició.

És a dir, amb unes tècniques molt senzilles podem extreure la posició de les mans amb molta precisió. Si no tinguéssim la imatge de profunditat, ens seria molt més difícil d'aconseguir el mateix resultat, per no dir impossible.



Imatge de profunditat obtinguda amb una Kinect. Les zones més clares representen més proximitat amb la càmera. Les zones més fosques representen més allunyament de la càmera. Fixeu-vos que la part del coll és fora del rang de profunditat i d'alguna manera no existeix en aquesta imatge.
Font: www.lawriecape.co.uk

L'altra característica impressionant de la Kinect és la capacitat per a reconèixer les postures i l'orientació d'una figura humana davant la càmera. El mateix maquinari de dins de la Kinect ens diu quantes persones detecta a la imatge i quina és la postura de cada una. Les dades que expressen una postura normalment són els angles de les articulacions. La Kinect divideix el cos en tres dimensions (cap, avantbraç, braç, tronc superior i inferior, cap, cuixes i cames).

D'aquesta capacitat han sorgit multitud de tècniques per a interactuar amb els usuaris. Per exemple, generant un model 3D en temps real que es mou tal com es mou l'usuari.

En aquesta imatge veiem com Kinect pinta una estructura d'esquelet simplificada que copia la posició tridimensional de l'usuari de davant la càmera.
Font: weblog.200ok.com.au.

3.5. Disseny d'interaccions

Un cop ja hem extret les dades de la imatge, hem d'utilitzar aquesta informació per a prendre decisions dins el nostre sistema.

Aquesta és l'etapa del procés en què, a partir de les dades extretes, podem prendre decisions i accions relacionades.

Exemple

Per exemple, si l'usuari ha mogut els braços, activarem una seqüència determinada d'operacions o, si la peça vermella és molt a prop de la blava, farem sonar una música.

Generalment, obtindrem unes dades numèriques (coordenades del centre de masses, quantitats de píxels d'un determinat color, velocitat de moviment...) que haurem de transformar i usar com a control de tot tipus d'esdeveniment. Òbviament, és en aquesta part que hem de ser creatius i analitzar quins són els mapatges més eficients per tal de dissenyar una interacció funcional i comprensible.

4. Més enllà. Recursos i bibliografia específica

4.1. Referències

Computer visions:
http://en.Wikipedia.org/wiki/Computer_vision

Processing Tutorials: Getting Started with Video Processing via OpenCV:
http://createdigitalmotion.com/2009/02/
processing-tutorials-getting-started-with-video-processing-via-opencv/

OpenCV Motion Tracking, Face Recognition with Processing: I'm Forever Popping Bubbles:
http://createdigitalmotion.com/2009/02/opencv-motion-tracking-face-recognition-with-processing-im-forever-popping-bubbles/

Processing OpenCV Tutorial 1:
http://andybest.net/2009/02/processing-opencv-tutorial-1/

Introduction to programming with OpenCV:
http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/opencv-intro.html

The Five Reactive Books:
http://www.youtube.com/watch?v=nA_UTUvC4h8

Project3 – New Interactions with Kinect and Computer Vision:
http://golancourses.net/2011spring/projects/project-3-interaction/

Microsoft explica como funciona Kinect:
http://materiageek.com/2011/03/microsoft-explica-como-funciona-kinect/

4.2. Bibliografia

Programming Interactivity: A Designer's Guide to Processing, Arduino, and openFramework:
http://www.amazon.com/Programming-Interactivity-Designers-Processing-Openframeworks/dp/0596154143