

Gràfics digitals

Jordi Alberich
Albert Corral
David Gómez Fontanills
Alba Ferrer Franquesa
Àlex Sánchez Vila

PID_00236888

Índex

1. Introducció als gràfics digitals	5
1.1. Imatge digital i imatge analògica	5
1.1.1. Introducció: la codificació digital binària	5
1.1.2. Del bit al caràcter alfanumèric	7
1.1.3. Del bit al píxel	8
2. Gràfics de mapa de bits	10
2.1. Una retícula de píxels	10
2.2. Resolució de la imatge	10
2.3. Resolució i interpolació	13
2.4. Càlcul de la resolució	14
2.4.1. Càlcul de la resolució per pantalla	16
2.4.2. Càlcul de la resolució per impressió digital	16
2.4.3. Càlcul de la resolució per impressió analògica o convencional	17
2.5. Resolució de pantalla	18
2.6. Profunditat de color	20
2.7. Modes de color	22
2.7.1. Mode de color en mapa de bits o monocromàtic	24
2.7.2. Mode de color indexat	25
2.7.3. Mode de color escala de grisos	26
2.7.4. Mode de color RGB	27
2.8. Sistemes classificadors de color i els modes de color	28
2.8.1. Codificació HSB	29
2.8.2. Codificació hexadecimal	30
2.8.3. Codificació CIE L*a*b	33
2.8.4. Biblioteques de color. Les tintes planes o directes	33
3. Gràfics vectorials	36
3.1. Estructuració en objectes vectorials	36
3.2. Escalabilitat	37
3.3. Arxius vectorials i els objectes Bézier	38
3.3.1. Objectes Bézier	38
3.3.2. Bézier en el programari gràfic	38
3.3.3. Treball de precisió amb les corbes de Bézier	40
3.4. Propietats dels objectes vectorials	41
3.4.1. Propietats de la línia o filet	42
3.4.2. Transformacions dels objectes vectorials	44
3.4.3. Composició de gràfics vectorials	45
4. Formats d'arxiu per a gràfics	47
4.1. Tipus de format	47

4.2.	Formats per a gràfics vectorials i metafitxers	48
4.2.1.	EPS, un primer format vectorial per a arts gràfiques	48
4.2.2.	PDF, un metaformat versàtil	49
4.2.3.	SWF, un format vectorial multimèdia	50
4.2.4.	SVG, un format vectorial obert per Internet	51
4.3.	Formats per a mapa de bits	51
4.3.1.	TIFF, el format d'impressió	51
4.3.2.	GIF, el format precursor per web	52
4.3.3.	JPEG, més enllà de GIF	53
4.3.4.	PNG, un format definitiu per a la xarxa?	55
4.3.5.	Transparència d'índex i transparència alfa	56
4.3.6.	Sistemes de compressió	57
4.3.7.	Recursos d'optimització pels formats GIF i PNG	59
4.3.8.	LZW i el desenvolupament del format PNG	63
4.4.	Programari i formats natius	64
4.5.	Taula informativa dels diferents formats	65
5.	Programari de creació i tractament gràfic.....	68
5.1.	Programari <i>bitmap</i> i vectorial	68
5.1.1.	Entorns de treball vectorial	69
5.1.2.	Entorns de treball <i>bitmap</i>	69
5.1.3.	Confluències	71
5.2.	Programari orientat a impressió i orientat al web	73
5.2.1.	Programari orientat a impressió	73
5.2.2.	Programari orientat al web	75
5.3.	Programari propietari i <i>open-source</i>	76
5.3.1.	El programari	76
5.3.2.	L'aparició del programari de propietat	77
5.3.3.	El projecte GNU	78
5.3.4.	La llicència GPL	78
5.3.5.	El moviment del programari lliure	78
5.3.6.	Programari lliure per a gràfics	79
Bibliografia.....	81	

1. Introducció als gràfics digitals

1.1. Imatge digital i imatge analògica

1.1.1. Introducció: la codificació digital binària

La digitalització de dades ha suposat una revolució en les possibilitats de manipulació i tractament de la informació, sense comparació en la història de la humanitat. De fet, ha estat la pròpia digitalització la que ha generat el concepte d'imatge digital en contraposició amb la imatge analògica clàssica. En realitat, analògica és tota aquella imatge que no és digital. És analògica perquè no ha estat generada mitjançant un procés codificat ni s'emmagatzema tampoc de forma codificada. Durant segles aquesta ha estat la tecnologia de reproducció dels gràfics.

Una imatge digital és un gràfic codificat per mitjà de dígit (símbols arbitraris que no guarden una analogia amb el referent que representen). Anteriorment l'home ja havia recorregut a la codificació per a la reproducció d'imatges. Així, per exemple, el teixit de jacquard usava targetes perforades per als telers. De fet, aquestes són l'antecedent directe de les targetes perforades que utilitzaran els primers ordinadors.

Sketchpad d'Ivan Sutherland

Els sistemes de disseny assistit per ordinador, també coneguts per les sigles angleses CAD, són sistemes de programari i, de vegades, maquinari molt avançats. Però fa molt de temps que rondem pel cap dels programadors, no són res de nou. Vet aquí l'avi dels CAD actuals, el sistema Sketchpad.

L'Sketchpad va ser desenvolupat l'any 1963 per Ivan Sutherland com a part de la seva tesi doctoral. Introduïa alguns avenços que trigarien a desenvolupar-se com a eines d'ús comú. Proposava, per exemple, la interfície gràfica d'usuari (GUI), dècades abans que s'estenguessin per tot el planeta informàtic.

El sistema Sketchpad es va crear al Laboratori Lincoln del MIT sobre un ordinador TX-2, una de les màquines més avançades d'aleshores, ja que tenia 320 Kb de memòria base i 8 MB de memòria externa en forma de cintes magnètiques. Quant a dispositius gràfics, estava dotat amb un monitor de set polzades 1024 × 1024, a més d'un punter òptic i un perifèric equivalent al que considerariem l'actual ratolí.



Sketchpad, un programari de dibuix de gràfics d'enginyeria per a l'ordinador TX-2. Creat el 1963 per Ivan Sutherland com a part del seu doctorat en el Massachusetts Institute of Technology. Va ser la primera interfície gràfica d'usuari per a ordinadors. Fotògraf desconegut. Aquesta imatge es reproduïx acollint-se al dret de citació o ressenya (art. 32 LPI), i està exclosa de la llicència per defecte d'aquests materials.

La codificació digital de la imatge es fonamenta en el sistema binari de les màquines que el generen. Aquesta codificació és possible gràcies a la diferenciació que es produeix, en tot sistema electrònic, entre el pas i el no pas d'electricitat durant una unitat de temps determinada. Així, les dues alternatives (encès o apagat) per una mateixa posició (la unitat de temps) són assimilades com a dos xifres diferenciades per la mateixa posició: el número zero (apagat o *off*) i el número 1 (encès o *on*). Aquesta unitat (amb dos valors possibles) és l'anomenat bit (de l'anglès *binary digit*) i constitueix el fonament, en definitiva, de tot el sistema digital.

Tanmateix, un bit únicament permet generar dos valors diferents per cada unitat. Per tant la possible informació que pot transmetre un bit és molt limitada. Per aquesta raó es decidí recórrer a la concatenació de diversos bits per disposar d'un ventall superior de valors. I és que la combinació successiva dels dos valors possibles, que permet cada bit d'aquesta cadena, n'augmenta exponencialment el rang de valors i, per tant, les possibilitats d'informació transmissible.

La indústria digital ha convingut a proposar una cadena de 8 bits com la unitat immediatament superior al bit. Aquesta unitat superior s'anomena *byte* i permet un rang de 256 possibilitats, resultants de la combinació de valors dels 8 bits binaris. L'articulació de diferents *bytes* permet generar, al seu torn, cadenes informatives superiors fins a construir un sofisticat sistema que permet generar, gestionar i comunicar qualsevol informació imaginable.

La codificació digital, d'altra banda, en basar-se en un simple sistema binari de zeros i uns, facilita la conservació (íntegra) de la informació en qualsevol suport capaç de reproduir zeros o uns. Aquest és el cas, per exemple, dels suports magnètics o òptics. Igualment aquesta simplicitat garanteix una reproducció (descodificació) i duplicació amb total fidelitat de la informació origi-

nal transmesa o reproduïda. Simplement es copien o comuniquen números. Per tant, ja no hi interfereix el possible «soroll» que sí que podíem trobar en la duplicació analògica.

Bit, la base del sistema binari

En informàtica, bit és la contracció de l'expressió *binari digit* (**dígit binari**). El bit representa una variable binària en què cada dígit pren el valor de 0 o el valor d'1. Per tant, dins de l'ordinador, constitueix la unitat més petita en què es pot codificar la informació (paraula, nombre, instrucció). En conseqüència, qualsevol informació es mesura pel nombre de bits necessaris per a la seva representació en un sistema binari.

El sistema binari treballa sobre la base 2, és a dir, únicament utilitza dues xifres per a expressar qualsevol quantitat. Aquestes xifres són els dos valors que pot tenir un bit: 0 i 1. A nivell electrònic la màquina considera:

1) **valor 0** = transmissió d'un senyal elèctric de tensió entre 0 volts i 0,0 volts.

2) **valor 1** = transmissió d'un senyal elèctric de tensió entre 0,8 volts i 4,5 volts.

Del bit, com a unitat, se'n deriven la resta d'unitats de mesura. Així obtenim el *byte*, cadena formada per una concatenació de 8 bits. Les unitats superiors successives són el *kilobyte*, el *megabyte*, el *gigabyte*, el *terabyte*... L'augment de l'escala es troba determinat per la potència de 2 més pròxima a 1000 que s'expressa segons la fórmula:

$$2^{10}=1.024$$

Cada unitat manté aquesta relació numèrica amb l'anterior però també amb la immediatament següent:

$$1 \text{ bit} = 0 \text{ o } 1$$

$$1 \text{ byte} = 8 \text{ bits}$$

$$1 \text{ kilobyte} = 1.024 \text{ bytes}$$

$$1 \text{ megabyte} = 1.024 \text{ kilobytes}$$

$$1 \text{ gigabyte} = 1.024 \text{ megabytes}$$

1.1.2. Del bit al caràcter alfanumèric

Tal com hem comentat, un *byte* (octet de bits) permet 256 possibilitats. Aquest *byte* (capacitat de memòria digital) pot associar-se a cada posició d'un caràcter de text. Cada unitat, doncs, d'una cadena tipogràfica (cada «espai») podrà reproduir fins a 256 possibilitats diferents per caràcter. Aquest rang és més que suficient per reproduir el codi alfanumèric de l'alfabet llatí (lletres, xifres, signes...). I, de fet, tal com veurem en abordar les tipografies digitals, tecnologies de fonts com ara TrueType o PS1 treballen a ple rendiment amb caràcters definits per un únic *byte*.

Tanmateix, cal saber que altres alfabetos com, per exemple, el xinès, el japonès o el coreà no en tenen prou amb les 256 possibilitats per posició que permet un únic *byte*. Necessiten més capacitat per reproduir tots els caràcters amb què compten els seus idiomes. La solució tècnica es troba en l'adjudicació d'un doble *byte* per cada caràcter. D'aquesta manera el rang de possibilitats augmenta fins a 65.536 possibilitats per caràcter. Aquesta major versatilitat, codificada mitjançant un sistema anomenat Unicode¹, permet associar un grafema (llettra, xifra, signe...) de qualsevol alfabet (i fins i tot, diferents alfabetos alhora) a

Possibilitats

$$1 \text{ bit: } 2^1 = 2$$

$$1 \text{ byte: } 2^8 = 256$$

$$2 \text{ bytes: } 2^{16} = 65,536$$

Vegeu també

Abordarem les tipografies digitals en l'apartat «3.Tipografia digital» del mòdul de «Tipografia i composició tipogràfica» de la nostra assignatura.

cada posició. Ara bé, només una tecnologia tipogràfica digital, posterior a PS1 i TrueType, com són les fonts OpenType adjudiquen un doble *byte* per cada caràcter.

⁽¹⁾Unicode és un estàndard per a la codificació digital de caràcters alfanumèrics i altres. Va ser ideat per al suport de caràcters en tots els idiomes del planeta, i per a la reproducció en sistemes digitals i a la xarxa. El codi és el resultat del consorci homònim format per empreses del sector com Apple, Microsoft, Adobe i d'altres, i institucions acadèmiques i professionals.

1.1.3. Del bit al píxel

En la creació i reproducció d'imatges, tot i que el sistema digital treballa igualment sobre codi binari, utilitza dues estructuracions diferenciades de la informació d'imatge. Aquests tipus d'arquitectura digital corresponen a dues classes d'imatge també diferents. Ens referim a la diferent codificació digital dels grafismes de mapa de bits i dels grafismes vectorials.

Tot i que més endavant abordarem les respectives diferències, per ara avançarem, per tal de facilitar l'explicació, que les imatges en mapa de bits es generen sobre la base d'una quadrícula d'elements continus però cromàticament independents. La visualització a distància del conjunt d'aquesta quadrícula genera la percepció d'una imatge cromàticament modulada (com ara una fotografia) i contínua (sense diferenciar els petits elements que formen la imatge). Cada element d'aquesta quadrícula és anomenat píxel i cada píxel pot reproduir un color diferent. La paleta cromàtica, però, que podrà reproduir cada píxel dependrà de la càrrega de memòria associada. És a dir, en funció de si un píxel compta amb una capacitat d'un bit, d'un *byte* o de tres *bytes* podrà reproduir més o menys colors. Aquesta associació entre càrrega de memòria i capacitat de reproducció cromàtica és coneguda com a profunditat de color d'una imatge.

Si un píxel només té associat un bit, únicament podrà reproduir dos valors cromàtics diferents. És aquest el mode de color propi de les imatges monocromàtiques (blanc o negre, sense escala de grisos). Pel contrari, quan una imatge proporciona a cada píxel un *byte* de memòria, aquest podrà reproduir 256 valors cromàtics diferents. Aquesta càrrega permet reproduir imatges en escala de grisos (conegudes pel gran públic com a «imatges en blanc i negre»). Aquestes utilitzen una progressió de grisos entre el blanc i el negre absoluts, tots dos inclosos, per tal de reproduir una imatge fotogràfica versemblant (tot i que acromàtica). No disposen de prou capacitat d'informació per reproduir imatges modulades cromàtiques com ara fotografies digitals a color.

Paleta indexada, color a 8 bits o 1 byte

Un format d'imatge digital com el GIF permet gestionar imatges en mapa de bits cromàtiques («a color») adjudicant només 1 *byte* per cada píxel. Aquestes 256 possibilitats no permeten reproduir tots els colors possibles, així que el format compta amb un recurs anomenat «paleta indexada». Aquesta paleta és una selecció o llista (d'aquí el concepte d'indexada) dels 256 colors, que considera més versàtils o habituals, per reproduir en cada píxel. El resultat són imatges cromàticament limitades (rang cromàtic forçat), però que certament presenten poc pes en memòria.

Les imatges cromàtiques («a color») en mapa de bits necessiten associar un mínim de 3 *bytes* (24 bits) a cada píxel. Aquestes són les imatges en mode RGB (*Red, Green, Blue*). Cadascun d'aquests tres canals de color primaris (vermell, verd i blau) compten amb un *byte* de memòria. Poden, per tant, reproduir 256 variacions cromàtiques en el seu canal. La combinació total del *byte* o 8 bits (256 possibilitats) associats a cada canal possibilita un rang cromàtic de 16,7 milions de colors (256 vermells x 256 verds x 256 blaus) a reproduir en cada píxel. Aquest mode cromàtic s'apropa de forma força versemblant a l'espai cromàtic perceptible (que seguirà essent, en tot cas, superior). Per tant, una imatge *bitmap* RGB (3 *bytes* per píxel) permet reproduir una fotografia digital a color.

2. Gràfics de mapa de bits

2.1. Una retícula de píxels

Els gràfics de mapa de bits construeixen la imatge segons la configuració d'una quadrícula regular i bidimensional (en alçada i amplada) d'elements continus i equivalents, on cadascun d'ells pot reproduir un valor cromàtic diferenciat. La visualització del conjunt d'aquests elements cromàtics (punts descriptius d'imatge), a certa distància, genera la percepció de la imatge. Cadascun d'aquests elements de la quadrícula, com ja hem comentat, són els anomenats píxels². Cada píxel reproduceix un color, que pot ser igual o diferent al del píxel contigu.

⁽²⁾ *Píxel* és un terme acrònim procedent del món anglosaxó, resultant de la contracció del concepte *picture element*. El seu significat podria ser element d'imatge.

1 píxel = 1 color

La imatge en mapa de bits es troba totalment determinada per dos paràmetres derivats de l'estructuració en píxels: la resolució i la profunditat de color.

El píxel, per constitució, no té mesura, és una unitat relativa que per ser significativa s'ha d'associar a una mesura concreta. La resolució és la relació d'elements de descripció d'imatge digital per unitat de superfície física o, el que és el mateix, el nombre de píxels per unitat de mesura que presenta un gràfic digital de mapa de bits.

I si la resolució és el nombre (relatiu) d'unitats descriptives d'una imatge de mapa de bits, la profunditat de to constitueix la càrrega de memòria adjudicada a cadascun d'aquests píxels i que determinarà, al seu torn, la seva capacitat per reproduir una gamma cromàtica més gran o més petita.

2.2. Resolució de la imatge

La **resolució d'una imatge** és la relació d'unitats de descripció d'imatge (píxels) presents per unitat de superfície física de la imatge (polzada o centímetre).

El píxel constitueix la unitat (mínima) de la imatge digital *bitmap*. Cada píxel reproduceix un valor cromàtic propi, però que sempre serà uniforme per a tot el píxel (no podem, doncs, reproduir un degradat o més d'un color alhora en un píxel). La visualització global del conjunt de píxels de la imatge (cadascun amb el seu color), a certa distància, permet la percepció de la imatge com un continu. Per tant, la reproducció de la imatge depèn del nombre de píxels

presentes i, per això és important la resolució. Com més quantitat de píxels per unitat de mesura, més difícil serà la seva distinció. Amb poca quantitat de píxels per mesura, resulta fàcil percebre l'estructura «pixelada».

Sovint fem en l'àmbit gràfic diversos termes equivalents per expressar el mateix concepte d'imatge en mapa de bits. D'entre tots cal destacar molt especialment el concepte d'imatge tramada, que sovint s'utilitza amb el seu equivalent en anglès: imatge *raster*. El concepte de *raster* al·ludeix a la quadrícula regular que constitueix la trama de píxels de la imatge.

imatge *bitmap* = imatge *raster*

Els píxels són unitats virtuals. No són unitats físiques absolutes, és a dir, no presenten pròpiament unes dimensions físiques definides. Un píxel no mesura una quantitat concreta en centímetres, mil·límetres o micres. Pel contrari, són elements que s'adapten a les dimensions físiques totals (alçada i amplada) de la imatge *bitmap*. De fet, podríem entendre que són «flexibles».

Exemple

Posem per cas que tenim una imatge definida per 1024 píxels d'alçada i 768 píxels d'amplada, que presenta unes dimensions físiques de 10,24 cm d'alçada i 7,68 cm d'amplada. La imatge, per tant, presenta una definició o resolució de 100 píxels per cada centímetre lineal.

Un cop codificada la imatge amb aquests paràmetres, si doblem les dimensions físiques de la imatge fins als 20,48 cm d'alçada i els 15,36 cm d'amplada, els píxels no augmenten, senzillament també es redimensionen. En el doble de distància continuem tenint el mateix nombre de píxels. Hem abaixat la resolució de la imatge. Podríem entendre que «s'eixemplen».

Per tant, el còmput global de píxels seguirà essent de 1024 píxels d'alçada i 768 píxels d'amplada. El que sí que es modificarà serà la densitat de píxels per unitat de superfície. La distribució inicial de 100 píxels per centímetre es reduirà, amb l'ampliació, a la meitat, fins als 50 píxels per centímetre. Aquesta relativa paradoxa resulta fonamental d'entendre per treballar amb resolucions d'imatge: quan ampliem al doble la mesura de la imatge, es redueix la seva resolució a la meitat.

La interrelació entre dimensions físiques i quantitat de píxels condiciona la densitat de píxels de tota imatge *bitmap*. El paràmetre que gestiona aquesta relació és l'anomenada resolució digital d'una imatge. Parametritza la quantitat de píxels presents per cada unitat física de la imatge. Tot i que fins ara hem utilitzat com a mesura els centímetres per tal d'explicar la resolució, en realitat, en el món gràfic acostumem a utilitzar, per influència anglosaxona, la polzada³ com a unitat de mesura física.

La resolució d'una imatge digital s'expressa en píxels per polzada (abreujada com a ppp), relació corresponent a l'expressió anglosaxona *pixels per inch* (ppi).

⁽³⁾Una polzada és una unitat de mesura física del sistema imperial britànic. El seu valor unitari correspon a 2,54 centímetres (o 25,4 mil·límetres) del sistema mètric decimal.

Píxels per polzada

L'expressió píxels per polzada es refereix a la **polzada lineal**. Així, per exemple:

Una imatge amb dimensions d'1 polzada (2,54 cm) d'alçada × 1 polzada (2,54 cm) d'amplada i una resolució de 300 ppp, estarà constituïda per 300 píxels d'alçada × 300 píxels d'amplada, és a dir, 90.000 píxels en total ($300 \times 300 = 90.000$).

Per tant, una imatge amb una resolució de 300 ppp presenta 90.000 píxels en cada polzada quadrada de la imatge.

Si aquesta imatge de 300 ppp de resolució presenta unes dimensions de 5 polzades (12,7 cm) x 3 polzades (7,62 cm), la quantitat total de píxels de la imatge serà de 1.350.000 píxels totals. Per a fer una comparativa homologable, si la càmera digital de fotografia obté imatges de 1500 x 900 píxels en el llenguatge de fotografia digital es parlaria d'1,3 megapíxels (1.350.000 píxels).

Els píxels d'una imatge no deixen de ser els punts descriptius generats durant la lectura òptica de la imatge, és a dir, durant la seva creació o captació fotogràfica. La quantitat global dependrà, doncs, de la capacitat de lectura (resolució) del dispositiu digitalitzador (generador de la imatge *bitmap*), és a dir, de l'escàner o del dispositiu digital fotogràfic (càmera, *smartphone*, *tablet*...) o, en el seu defecte, del valor de resolució que determinem en la creació de l'arxiu en un programa de gràfics *raster* com ara Adobe Photoshop.

Aquest total de píxels es distribuïran de forma contínua i uniforme (en estructura de quadrícula) per tota l'àrea de la imatge. Podríem entendre, doncs, que l'alçada i amplada físiques determinen el perímetre de la imatge. Un cop determinades aquestes mesures (perímetre i número de píxels), sabrem quina resolució presenta la imatge. Aquest nombre total de píxels és inamovible un cop configurat. Si ampliem o reduïm físicament la imatge, el nombre de píxels roman constant i, per tant, guanyarem o perdrem resolució.

La guanyarem si reduïm les dimensions o la perdrem si ampliem les dimensions de la imatge.

Seguint aquesta lògica, podem entendre que tota imatge en mapa de bits no és «escalable» o, el que és el mateix, ampliable sense límits perquè correlativament perdrem qualitat. A mesura que estirem la imatge pels seus vèrtexs, els píxels augmenten espacialment de manera proporcional. I, com possiblement ja sabem per la nostra experiència d'usuari, aquest increment forçat dels píxels, arribat un punt, genera l'anomenada «pixelació» de la imatge. Els píxels esdevenen distingibles com a quadrats cromatitzats i els contorns de les formes apareixen «dentats». Es trenca, en definitiva, la il·lusió de continuïtat de la imatge. Sobrepassat aquest valor, la imatge resultarà progressivament menys reconeixible.

Per tant, existeix un topall perceptiu, determinat per una quantitat mínima de píxels per polzada, que no convé superar per tal que la imatge no aparegui «pixelada». Aquesta resolució mínima, tanmateix, no és absoluta. Varia en funció del dispositiu de reproducció de la imatge.

Exemple

Una imatge *bitmap* per pantalla necessita una resolució mínima de 72 píxels per polzada (72 ppp), mentre que per una impressió digital convindria comptar amb una resolució al voltant d'uns 300 píxels per polzada (300 ppp).

Com més píxels presenti cada polzada de superfície de la imatge, major serà la definició cromàtica i també la nitidesa; ja que els punts també seran més petits i, per tant, menys visibles.

2.3. Resolució i interpolació

Cal aclarir que una vegada obtingut o generat el gràfic de mapa de bits (la fotografia), ja no podem augmentar-ne el nombre de píxels que el conformen. Aquesta operació d'increment no és possible, llevat que recorrem a la tècnica de la interpolació o remostreig.

La interpolació o remostreig consisteix en la generació artificial de nous píxels a partir de la clonació dels originals de la imatge. Aquesta tècnica permet augmentar la resolució i salvar, així, una eventual pixelació en la reproducció. Tanmateix, aquests píxels clonats ocasionen una pèrdua de nitidesa en la imatge que, segons el grau d'interpolació, resultarà apreciable en forma de desenfocament⁴.

⁽⁴⁾Per aquest motiu, tot i que les versions actuals dels programaris de tractament gràfic i els motors de processament dels RIP per als plòters més professionals ofereixen una capacitat d'interpolació molt sofisticada, no podem recomanar aquest mètode per a l'augment de la resolució d'una imatge, sense advertir dels perills del resultat imprès.

Podem redimensionar la imatge i els píxels incrementaran o reduiran proporcionalment les seves mides per tal d'emplenar les noves dimensions de la imatge. Però la quantitat total de píxels romandrà sempre estable. Allò que variarà, per tant, serà la resolució, és a dir, el nombre de píxels per unitat de superfície. Així doncs, dimensions i resolució d'imatge es troben interrelacionades de manera inversament proporcional.

Exemple

Podem observar com, en duplicar les dimensions físiques del gràfic digital següent (236 × 591 píxels), automàticament la resolució disminuirà de manera inversament proporcional (es reduirà a la meitat), i a la inversa. En tots els casos la quantitat total de píxels i la càrrega de memòria corresponent no variaran.

Quantitat de píxels		Dimensions físiques		Resolució (ppp, píxels per polzada)	Profunditat de color	Pes en memòria digital
Amplada	Alçària	Amplada	Alçària			
236 px	591 px	2 cm	5 cm	300 ppp	3 bytes	408,6 Kb
236 px	591 px	4 cm	10 cm	150 ppp	3 bytes	408,6 Kb
236 px	591 px	8 cm	20 cm	75 ppp	3 bytes	408,6 Kb

2.4. Càlcul de la resolució

Evitar la pixelació suposa un requisit ineludible en la reproducció de tota imatge, ja sigui per pantalla o per impressió. Aquest objectiu, tanmateix, resulta més crític en la impressió, atès que caldrà atribuir a la imatge digital resolucions diferenciades en funció dels dispositius d'impressió.

En la producció gràfica orientada a impressió intervenen diversos paràmetres que comparteixen el terme de resolució. Convé saber diferenciar-los:

1) Resolució digital o de digitalització: relació de píxels per polzada resultant de la lectura-captació d'una imatge per part d'un dispositiu de digitalització com ara l'escàner o un dispositiu fotogràfic digital.

Aquesta resolució, per tant, serà amb la qual la imatge digital «entrarà» en el flux digital de treball i/o reproducció i amb la qual treballaríem en el seu tractament mitjançant una aplicació per a l'edició gràfica (com ara Adobe Photoshop o GIMP⁵).

⁽⁵⁾ Les sigles GIMP corresponen al programa d'edició d'imatges en mapa de bits GNU Image Manipulation Programme. Aquesta és una aplicació de codi obert, lliure i accessible, dedicada al tractament d'imatges *raster*.

La unitat de resolució d'aquest paràmetre són els píxels per polzada (ppp).

2) Resolució de filmació: paràmetre relativament complex i relacionat amb la impressió convencional (òfset, flexografia o rotogravat). Per tant, no intervé en la impressió digital. Fonamentalment parametriza la capacitat d'exposició de punts de filmació (coneguts com a *spots*) per polzada d'un dispositiu de filmació de fotolits o de formes impressores.

Aquesta resolució determinarà, en definitiva, si és possible o no filmar sobre la forma impressora corresponent la lineatura d'impressió final desitjada.

La unitat de resolució d'aquest paràmetre és l'expressió anglosaxona dpi (*dots per inch*, és a dir, punts per polzada).

3) Lineatura d'impressió: relació de punts d'impressió per unitat mètrica. Com major sigui el nombre de línies de punts per polzada o centímetre, major quantitat de punts d'impressió obtindrem. Atès que la unitat de superfície

no varia (la polzada o el centímetre), en incrementar la lineatura i, per tant, el nombre de punts associats, lògicament aquests seran més petits (diàmetre menor). La impressió guanyarà en detall, definició i rang cromàtic.

El valor de lineatura, però, no es pot incrementar indefinidament. Presenta una lineatura màxima reproduïble, limitada tant pel propi dispositiu d'impressió (màquina i tipus de suport) com per la capacitat d'exposició de la filmadora associada (resolució de filmació, dpi).

La lineatura es mesura mitjançant la unitat lpp (línies per polzada) o lpcm (línies per centímetre). 150 lpp corresponen a 60 lpcm.

La definició, en conclusió, d'una imatge digital està directament relacionada amb la resolució, ja que com major sigui la quantitat de píxels per polzada que presenti, major precisió i rang cromàtic presentarà la imatge. Ara bé, cal tenir en compte que com més píxels tingui una imatge, major serà també el seu pes en memòria i més crítica la impressió. I és que les altes resolucions només són viables en impressió amb dispositius tecnològicament sofisticats.

D'altra banda, tot i que hem presentat la resolució mínima d'un dispositiu com el topall per sota del qual la imatge apareix pixelada, és igualment cert que, per sobre d'aquesta resolució, la possible millora visual progressivament serà menys apreciable. Per tant, una «sobreresolució» carrega el pes en memòria, podríem dir que de manera innecessària.

Major resolució (ppp) = major definició i rang cromàtic = major pes en memòria (MB)

En la digitalització d'un original (per escaneig) o directament per captació de la realitat (fotogràficament) convindria precisar la quantitat de píxels de lectura (resolució òptica). Aquest valor hauria de dependre de les dimensions físiques finals de la imatge a reproduir. Però no sempre és possible (o viable) configurar aquest valor (per exemple en els escàners), i quan ho és (com en els dispositius fotogràfics digitals) no sempre resulta prou aclaridor de la relació final que obtindrem entre dimensions i resolució.

Càmeres digitals, *smartphones* i megapíxels

Tant les càmeres digitals com els *smartphones* actuals informen de la seva resolució òptica mitjançant un valor anomenat megapíxel (MP). Aquesta unitat equival a 1 milió de píxels i indica (de forma arrodonida) el resultat màxim que un dispositiu pot fotografiar. El valor resultant sorgeix de la multiplicació dels píxels d'alçada pels píxels d'amplada.

Els escàners permeten configurar la resolució de digitalització de la nova imatge. Aquesta resolució, però, no té per què ser la màxima possible del dispositiu. Tal com hem explicat, com més resolució més pes en memòria, i no sempre serà necessària tanta informació.

La qüestió cabdal, doncs, és determinar la resolució digital exacta per una captació i gestió posterior eficient de la imatge digitalitzada. No ha de resultar excessiva, pel corresponent sobrepès en memòria, ni tampoc insuficient, per tal d'evitar la pixelació en la reproducció posterior. De fet, la solució no es troba tant en l'original a reproduir ni en l'escàner com en les característiques del dispositiu de sortida. Com hem indicat, cada dispositiu necessita d'una resolució mínima i un valor màxim per sobre del qual el guany visual no és tècnicament reproduïble i/o visualitzable de manera significativa.

La pràctica gràfica utilitza, en aquest sentit, tres mètodes de càlcul genèric que poden resultar-nos de gran ajuda per establir la resolució digital apropiada en funció de la sortida corresponent:

- 1) per pantalla
- 2) per impressió digital
- 3) per impressió analògica o convencional

2.4.1. Càlcul de la resolució per pantalla

Aquestes seran les imatges dirigides a la reproducció per pantalla (web, multimèdia, etc.).

$$\text{Resolució digital} = 72 \text{ ppp} \times \text{Factor de reproducció}$$

Factor de reproducció

El factor de reproducció (FR) és una variable que expressa la relació de redimensionament d'una imatge. El valor s'obté de la divisió d'una de les dimensions de la reproducció (l'alçada o l'amplada) entre la dimensió equivalent de l'original reproduït.

$$\text{Factor de reproducció} = \text{alçada final} / \text{alçada original} \text{ (o amplada final} / \text{amplada original)}$$

Aquesta fórmula es pot calcular indistintament per un costat o l'altre de la imatge, ja que cal redimensionar proporcionalment per evitar la distorsió d'una imatge. Així, el resultat serà sempre el mateix per als dos eixos. Per exemple, un original de 3 × 5 cm que es reproduïx a 6 × 10 cm presentarà un FR de 2 ($\text{FR} = 6/3 = 10/5 = 2$). Si les imatges es reproduïxen amb la mateixa mida que l'original, l'FR serà sempre 1 i per tant la seva incidència sobre el càlcul de la resolució serà innòcua.

El valor de 72 ppp, tot i la seva correcció, és un valor superat pels dispositius digitals actuals. Si volem explotar la definició de les pantalles d'aquests aparells caldria, doncs, augmentar també el nostre valor de càlcul.

2.4.2. Càlcul de la resolució per impressió digital

$$\text{Resolució digital} = 300 \text{ ppp} \times \text{FR}$$

PPP

Recordem que la unitat ppp significa píxels per polzada. És una unitat de resolució digital.

El valor de 300 ppp constitueix una dada genèrica referencial per a facilitar l'aprenentatge. Tot i resultar un valor correcte i suficient, els diferents dispositius digitals actuals ofereixen un rang de resolucions superior, amb valors màxims diverses vegades superiors a l'indicat.

2.4.3. Càlcul de la resolució per impressió analògica o convencional

En l'àmbit de la **producció impresa**, la resolució digital es determina en funció del valor de la lineatura d'impressió. Aquesta, al seu torn, es troba condicionada pels següents factors:

- 1) dimensions finals de la imatge en la impressió final (tenint en compte el factor de reproducció);
- 2) tipus de sistema d'impressió convencional, ja sigui òfset, rotogravat, flexografia o serigrafia;
- 3) naturalesa superficial del suport d'impressió final (paper, plàstic...).

$$\text{Resolució digital} = \text{Lineatura d'impressió} \times \text{Factor de qualitat} \times \text{FR}$$

La lineatura d'impressió o freqüència de trama constitueix la relació de punts de semitò per unitat de superfície que és capaç de reproduir un sistema d'impressió, en funció del suport concret utilitzat. Així doncs, varia per a cada sistema d'impressió i suport utilitzat. Aquesta relació es mesura mitjançant la unitat línies per polzada (lpp).

El factor de qualitat és una constant matemàtica que, tot i les variacions que proposen diferents autors, en funció del tipus d'imatge a reproduir, podem establir de forma genèrica en el valor numèric de 2.

Exemple

Cal imprimir una imatge amb les mateixes dimensions de l'original (FR = 1) en una impressió estàndard en òfset sobre paper òfset que treballa amb una lineatura de 150 lpp (línies per polzada o *lines per inch*). Del càlcul corresponent establím una resolució digital de 300 ppp per a l'original que hem d'imprimir:

$$\text{Resolució} = \text{Lineatura} \times \text{Factor de reproducció} \times \text{Factor de qualitat}$$

$$\text{Resolució} = 150 \text{ lpp} \times 1 \times 2 = 300 \text{ ppp}$$

Aquest valor de resolució serà el que assignarem a la imatge *bitmap* corresponent en el seu tractament gràfic en una aplicació com ara Adobe Photoshop (Imagen / Tamaño de imagen).

Sistema d'impressió	Trama	Factor de qualitat	Resolució imatge
Impressió digital		× 2	600 ppp

Factor de reproducció

Aquest factor resulta del quocient entre les respectives dimensions finals i les de l'original. Per exemple:

$$\text{FR} = 10 \text{ cm} / 5 \text{ cm} = 2$$

(o el que és el mateix, la imatge es reproduirà al 200% o el doble de gran).

Sistema d'impressió	Trama	Factor de qualitat	Resolució imatge
Òfset (paper òfset)	150 lpp	× 2	300 ppp

Tenint en compte, doncs, les diverses fórmules plantejades, i incorporant-hi el valor constant de resolució de 1200 ppp per a la impressió d'imatges monocromàtiques en qualsevol sistema, proposem la taula següent de resolucions (a tall introductori):

Tipus d'imatge	Dispositiu de sortida	Lineatura o freqüència de trama		Resolució digital*
Imatge monocromàtica (1 bit per píxel)	Pantalla	–		72 ppp
	Impressió digital i analògica	–		1.200 ppp
Imatge en escala de grisos (1 bytes per píxel) i imatge RGB (3 bytes per píxel)	Pantalla	–		72 ppp
	Impressió digital	–		300 ppp
	Impressió analògica	Òfset	85 a 175 lpp**	170 a 350 ppp
		Rotogravat	200 lpp	400 ppp
		Flexografia	75 a 150 lpp	150 a 300 ppp
Serigrafia		50 a 100 lpp	100 a 200 ppp	

* Tots els valors resolutius es reproduïxen amb un factor de reproducció 1, és a dir, sense ampliació ni reducció.

** Atès el caràcter introductori d'aquesta taula, no tenim en compte la lineatura per impressió en l'anomenat òfset sec.

Nota

Encara podríem desenvolupar més aquesta taula, detallant les diferents resolucions en funció dels materials concrets per a cada sistema d'impressió i incorporant-hi un altre valor fonamental que cal tenir en compte com és la resolució del dispositiu de filmació. La complexitat resultant, però, superaria abastament l'àmbit d'aquest mòdul.

2.5. Resolució de pantalla

La resolució de pantalla determina la qualitat de les imatges que es reproduïxen en el monitor o *display* d'un dispositiu digital.

Les pantalles vectorials van ser les estàndards durant les dècades dels anys seixanta i setanta. Representaven la imatge per mitjà d'un oscil·loscopi que movia un feix d'electrons i dibuixava la imatge en pantalla. Al principi dels vuitanta es van abandonar en favor de les pantalles d'escombratge basades en una retícula de píxels. I serà durant la dècada dels noranta que s'introduiran progressivament les actuals pantalles planes de plasma i cristall líquid basades també en una retícula de píxels. Aquest condicionament tecnològic és el que obliga, tant a imatges de mapa de bits com a imatges vectorials, a reproduir-se finalment sobre una pantalla basada en una retícula de píxels (*bitmap*).

La resolució d'una pantalla es determina per la quantitat de píxels que pot representar en la pròpia pantalla del dispositiu.

Durant molts anys la resolució dels monitors va ser limitada. Els ordinadors Macintosh d'Apple comptaven amb pantalles que representaven resolucions de només 72 píxels per polzada. La resta d'ordinadors personals –clònics compatibles amb el PC d'IBM– usaven pantalles que representaven 96 píxels per polzada. Així, la mida física d'una imatge digital en pantalla podia ser relativament estable i es va establir com un estàndard *de facto* l'anomenada resolució de pantalla a 72 ppp o 96 ppp. De fet, com hem vist anteriorment, encara prenem com a referència el valor de 72 ppp en l'autoedició d'imatges *bitmap* per pantalla. Tanmateix, el desenvolupament tecnològic actual amb l'aparició de nous dispositius ha permès superar de llarg la resolució de 72 ppp.

Actualment la resolució de pantalla és un valor variable que depèn del monitor que utilitzem. Hi ha un límit en la mida física del píxel en pantalla que depèn de la mida dels punts de fòsfor (en pantalles de rajos catòdics) o de la cel·la (en pantalles de cristall líquid). Aquest límit, però, depèn de la definició del monitor.

En conseqüència, no existeix una relació estable entre la resolució de la imatge digital i la mida física en la qual aquesta imatge es mostrarà en pantalla.

Una imatge digital es mostrarà amb la mateixa mida en qualsevol navegador web independentment de quina resolució hàgim establert (per exemple, 72 o 300 ppp). L'explicació rau en el fet que el monitor representa la imatge en funció de les seves dimensions (píxels d'amplada × píxels d'altura) i no de la resolució. De fet, la resolució decisiva és l'escollida en el nostre sistema operatiu, com també la mida física de la pantalla del monitor (en polzades) on visualitzem aquesta imatge.

Exemple de resolució de pantalla

Una imatge de 800×600 píxels es mostrarà a tota la pantalla si hem configurat 800×600 com a resolució en les preferències del sistema operatiu. I això serà així tant en una pantalla de 14", de 17" o de 19". La diferència és que la imatge es veurà físicament més gran en la pantalla de 19", ja que cada píxel serà també més gran (ocuparà més espai en la superfície de la pantalla). Si augmentem la resolució de 800×600 a 1.024×768 píxels, o més, aquesta imatge de 800×600 ja no emplenarà tota la pantalla, sinó una part.

Per tant, el que és important en les imatges que es mostren per pantalla no és la resolució (ppp) sinó la mida (píxels d'altura per píxels d'amplada), ja que amb més píxels es poden representar o reproduir millor els detalls de qualsevol imatge. Tot i així, el dissenyador no pot controlar totalment les dimensions

físiques en les quals es mostrarà aquesta mida en píxels. És l'usuari qui, per mitjà de la configuració de l'equip, té el control final de la seva experiència perceptiva.

2.6. Profunditat de color

Cada píxel, tal com hem remarcat en diverses ocasions, reproduceix un valor cromàtic propi i uniforme per a tot el seu espai. El rang cromàtic que pot reproduir (i per tant l'espai de color possible per aquest píxel en concret i per tota la imatge en general) dependrà del nombre de bits (o *bytes*) de memòria adjudicats a cada píxel.

Per tant, si la capacitat cromàtica d'un píxel depèn del nombre de bits que hi associem, com més càrrega de memòria hi adjudiquem, major serà la versatilitat cromàtica a reproduir. Ara bé, també serà major l'espai de memòria.

La càrrega de memòria associada a cada píxel determina, doncs, la capacitat de reproducció cromàtica de cada píxel i, per extensió, de la imatge. Aquest paràmetre és conegut com a profunditat de color del píxel o profunditat del píxel.

Convé insistir en què la càrrega de memòria i la corresponent profunditat de color pot variar. Però, sigui quina sigui, sempre serà la mateixa per a tota la imatge. És a dir, tots els píxels de la imatge tindran sempre la mateixa càrrega de memòria associada i per tant de profunditat de color. No és possible, doncs, que una imatge presenti alhora diferents profunditats de color o modes de color.

L'adjudicació de memòria a cada píxel (i per tant a la imatge) no és aleatòria. Els programes de tractament per gràfics en mapa de bits (com Adobe Photoshop) faciliten un ventall molt determinat de càrregues de memòria a assignar per píxel, que al seu torn suposaran uns espais de color també concrets.

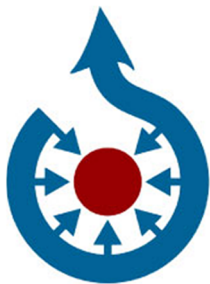
Mostrem a continuació una taula de les possibles codificacions de color per píxel en relació amb la memòria assignada, és a dir, una taula de la possible profunditat de color configurable.

Profunditat de color			
Càrrega de memòria per píxel	Gamma cromàtica reproducible	Mode de color (paleta cromàtica)	Tipus d'imatge
1 bit	$2^1 = 2$ tons	monocromàtic (anomenat mapa de bits en Photoshop)	Imatges planes (no modulades), cromàticament binàries, en blanc i negre (sense transparències ni matisos).

Profunditat de color			
Càrrega de memòria per píxel	Gamma cromàtica reproducible	Mode de color (paleta cromàtica)	Tipus d'imatge
8 bits (1 bytes)	$2^8 = 256$ tons	escala de grisos	Imatges modulades (fotogràfiques) en escala de grisos (conegudes com a imatges en b/n).
		paleta indexada	Imatges planes o modulades però amb una reproducció cromàtica limitada únicament a 256 tonalitats (generalment apreciable). Són les pròpies del format .gif.
24 bits (3 bytes)	$2^{24} = 16.777.216$ tons	RGB (color) 3 canals = 8 bits per canal	Imatges modulades (fotogràfiques) percebudes com a cromàticament reals.
32 bits (4 bytes)	$2^{32} = 4.294.967.296$ tons	CMYK 4 canals = 8 bits per canal	Imatges modulades a color, estructurades virtualment en les 4 separacions corresponents a la impressió quadricròmica.

Com veiem, la profunditat de color es troba proporcionalment relacionada amb el pes en memòria. Convé tenir en compte aquesta relació en la determinació de la profunditat de les imatges, especialment per aquelles que s'han de reproduir a la xarxa; tot i que, certament, la potència de les xarxes i dels dispositius actuals relativitzen la importància d'aquesta consideració.

Si la resolució parametriza la quantitat (relativa) de píxels i cadascun d'aquests píxels reproduceix un valor cromàtic, en funció de la profunditat de to assignada, podem convenir que com més píxels trobem per a les mateixes dimensions físiques, necessàriament aquells hauran de resultar menors (i per tant menys apreciables) i, en conseqüència, la imatge es reproduirà amb més detall i amb un rang cromàtic més ampli, tot i que també el pes en memòria serà correlativament superior.



N'hi ha prou amb 8 bits per a representar aquesta icona que identifica el repositori d'imatges i arxius lliures del projecte "Commons" de la Fundació Wikimedia en què s'utilitzen colors plans.

© Fundació Wikimedia. Aquesta imatge es reproduceix acollint-se al dret de citació o ressenya (art.32 LPI) i s'exclou de la llicència per defecte d'aquests materials.



No n'hi ha prou amb 8 bits per a aquesta fotografia.

© Chris Brown (zoonabar) 2010 - UOC 2010. Creative Commons Reconeixement CompartirIgual 3.0. Publicada originàriament a Flickr.



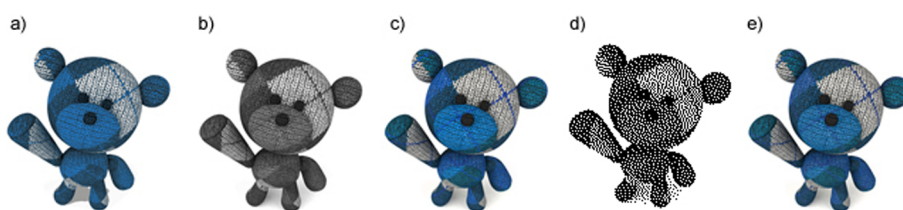
N'hi ha prou amb 24 bits per a representar aquesta fotografia amb bona qualitat.

© Chris Brown (zoonabar) 2010. Creative Commons Reconeixement Compartir Igual 2.0. Publicada originàriament a Flickr.

2.7. Modes de color

Un programa de tractament de gràfics en mapa de bits (com Adobe Photoshop) ofereix els següents modes de color lligats a la profunditat de color corresponent:

Mode de color	Profunditat de color / càrrega de memòria	Canals
Mapa de bits	1 bit per píxel	1
Escala de grisos	8 bits per píxel	1
Monotò / Duotò...	8 bits per píxel	1
Color indexat	8 bits per píxel	1
RGB	24 bits per píxel	3
CMYK	32 bits per píxel	4
Lab	24 bits per píxel	3
Multicanal	8 bits per píxel	1



a) Indexat. 8 bits. 14 Kb. b) Escala de grisos. 8 bits. 23 Kb. c) RGB. 24 bits. 45 Kb. d) B/N. 1 bit. 12 Kb. e) CMYK. 32 bits. 63 Kb.

En un programa com Photoshop el mode de color triat no deixa de ser la selecció cromàtica disponible i reproduïble. Els diferents modes reproduïxen el color mitjançant els anomenats canals. Cada canal representa el rang complet de cadascun dels colors primaris. Si la selecció és acromàtica (com en els modes escala de grisos o l'anomenat «mapa de bits» a Photoshop) el mode només utilitza un únic canal («de blanc a negre»). Si treballem en un mode cromàtic com RGB, aquest utilitza tres canals (un per cada color primari). En tot cas, el mode de color seleccionat assigna al canal una càrrega de memòria concreta per píxel. Recordem que aquest pes en memòria permet reproduir més o menys colors per píxel.

Com hem comentat, per exemple, el mode RGB utilitza tres canals (*Red*, *Green* i *Blue*), on cadascun pot reproduir 256 tonalitats de modulacions. Així, compta amb 256 possibles vermells, 256 possibles verds i 256 possibles blaus. El color final serà el resultat de la combinació d'aquests tres canals. Per tant, RGB pot reproduir fins a 16,7 milions de tonalitats.

Els diferents sistemes de codificació del color que es poden aplicar al document en què treballem són coneguts com a modes de color.

2.7.1. Mode de color en mapa de bits o monocromàtic

El mode en mapa de bits s'anomena també monocromàtic perquè només reproduïx dues tonalitats: blanc i negre. No reproduïx, doncs, grisos, ni tonalitats ni tampoc transparències. Aquesta limitació s'explica perquè només assigna 1 bit (dues possibilitats) per píxel.

Aquest mode genera imatges en mapa de bits però sense modulacions. Ofereixen, doncs, imatges planes, semblants a les il·lustracions vectorials, anomenades també en l'argot gràfic imatges de línia (o plomes).

Necessiten una resolució digital molt superior a la desitjable per una imatge modulada (fotografia) per impressió digital o convencional. Altrament la imatge mostrarà (de manera molt evident) pixelació (efecte dentat) en la separació entre píxels blancs i negres.

Resolució digital per impressió digital o analògica òfset (FR = 1)

Imatge en mapa de bits o monocromàtica (1 bit/píxel)	Imatge modulada RGB (24 bits/píxel)
1200 ppp	300 ppp

Tot i la resolució molt superior, com que la profunditat de color és mínima (1 bit per píxel) la imatge pesarà poc. De fet, el seu pes, respectant les mateixes dimensions i resolució de 300 ppp, serà entorn a 24 vegades menor que si la mateixa imatge es trobés en mode de color RGB. Atès que el programa ens ho permet, i és desitjable, configurarem la resolució de la imatge en mapa de bits a 1200 ppp. Tot i així, mantenint les mateixes dimensions, la imatge encara pesa menys que si la convertim a RGB.

Mode de color	Imatge en mapa de bits o monocromàtica		Imatge modulada RGB
Dimensions físiques	15 x 11 cm		
Profunditat de píxel	1 bit/píxel		24 bits/píxel
Resolució digital	300 ppp	1200 ppp	300 ppp
Pes en memòria digital	282 Kb	4,39 MB	6,59 MB

Altres programes, com ara GIMP, no faciliten una conversió automàtica a aquest mode de color. Per aconseguir-ho cal determinar una profunditat de només 1 bit en convertir la imatge a mode indexat.

Imatge original



a)



b)



a) Imatge monocromàtica (des de Photoshop, mode «Mapa de bits / Método 50% de umbral».)
b) Imatge monocromàtica (indexació des de GIMP usant l'opció "Usar paleta en blanco y negro (1-bit)").

2.7.2. Mode de color indexat

És un mode de color que treballa amb un sol canal de 8 bits. Per tant, pot representar un màxim de 256 valors, que reproduiran una paleta cromàtica concreta d'igual nombre de tonalitats.

Destinat a la utilització per al web o a produccions multimèdia que s'hauran de veure en pantalla. El fet de reduir el nombre de bits en la codificació de la imatge en redueix el pes (memòria que ocuparà l'arxiu). Es poden fer servir paletes amb menys de 256 colors, amb la qual cosa també reduïrem el pes.

La majoria de programes de gràfics permeten escollir el nombre de colors de la paleta (o el nombre de bits per a codificar-lo). Això es pot fer en escollir el mode de color en el qual es treballarà o bé en generar el gràfic corresponent per a la publicació web. També permeten seleccionar tipus de tramats que ajudaran a obtenir colors fora de la paleta indexada en el gràfic per mitjà de la barreja partitiva.

a)



b)



a) Imatge indexada amb paleta «Web216 sense tramat». Els colors originals han estat substituïts pels colors més pròxims de la paleta.

b) Imatge indexada amb paleta «Web216» des de GIMP usant el mètode Floyd Steinberg. Els colors originals es reproduïxen per simulació (barreja partitiva) distribuïent de manera estocàstica els punts cromàtics de la paleta indexada.



Paleta 256 colors



Paleta 128 colors



Paleta 64 colors



Paleta 32 colors



Paleta 8 colors



Paleta 4 colors

En aquest exemple veiem com pot canviar una imatge tramada en funció del nombre de colors que tingui la paleta a la qual s'indexa.

© Alba Ferrer 2009 - Creative Commons Reconocimiento Compartir Igual 3.0.

2.7.3. Mode de color escala de grisos

El mode d'escala de grisos reproduïx qualsevol imatge modulada però de forma acromàtica. Aquestes imatges corresponen al que els usuaris anomenen imatges en blanc i negre. Reproduïx, doncs, diferents gradacions entre el blanc i el negre per reproduir l'escena corresponent. Aquest mètode també utilitza un únic canal de 8 bits que reproduirà fins a 256 valors de gris.

Imatge en escala de grisos



2.7.4. Mode de color RGB

RGB treballa a tres canals de 8 bits cadascun. Per tant, la imatge en mapa de bits compta amb 24 bits per píxel i pot representar fins a 16,7 milions de colors. Aquest és el mode de color digital i de pantalla que pot reproduir amb versemblança imatges modulades (fotogràfiques) a color.

Constitueix, de fet, el mode de color estàndard en autoedició⁶ ja que permet el màxim rang cromàtic possible per imatges fotogràfiques a color. És un mode de rang molt superior a la gamma cromàtica reproduïble per la impressió quadricròmica (CMYK), tot i que també existeixen tonalitats CMYK no reproduïbles exactament pel mode RGB. Treballar en aquest mode permet reproduir el mateix mode que utilitzaran els dispositius digitals de pantalla i també els de digitalització com ara escàners o dispositius fotogràfics (càmeres, *smartphones*...). Per tant, ajustant la gestió de color, podríem dissenyar els gràfics, tal com es visualitzarien en la posterior sortida web, per exemple.

⁶Entenem com a «autoedició» el conjunt de processos digitals, desenvolupats mitjançant els programes corresponents, de tractament d'originals, disseny, maquetació i generació de l'art final. El terme pròpiament volia reflectir (i així va ser concebut des d'una perspectiva de màrqueting) la capacitat de realitzar tots aquests processos des d'una mateixa màquina, l'estació de treball digital (l'ordinador).

Imatge en RGB



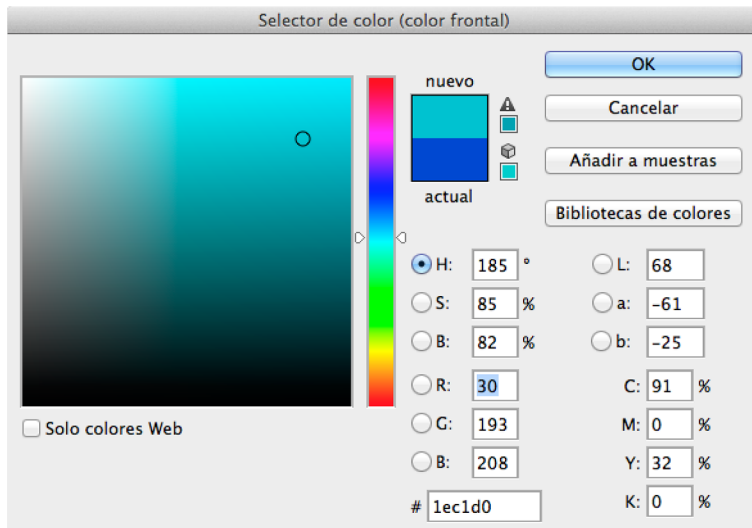
2.8. Sistemes classificadors de color i els modes de color

Una aplicació com Adobe Photoshop ens ofereix un recurs com és el «Selector de color» per tal de seleccionar o configurar la tonalitat concreta que vulguem aplicar a qualsevol element de la imatge *bitmap*.

Cal tenir clar que el mode de color (referit a l'estructuració cromàtica de la imatge) no és el mateix que el sistema classificador de color (el color a seleccionar en el programa).

La interfície d'aquesta eina permet triar el color en funció de diversos mètodes de codificació de color, quan no ho fem directament clicant sobre la representació de la gama cromàtica. Aquests sistemes de classificació cromàtica són HSB, RGB, color hexadecimal, Lab i CMYK. Fins i tot, podem accedir a través de la subopció «Biblioteca de colors» a les cartes de color de diferents sistemes de tintes planes com ara Pantone.

Selector de Color d'Adobe Photoshop



Aquesta imatge es reproduïx acollint-se al dret de citació o ressenya (art.32 LPI) i s'exclou de la llicència per defecte d'aquests materials.

Segui quin sigui el sistema que utilitzem, en seleccionar un color, automàticament aquest es recalculà en els altres sistemes, representant valors equivalents. Cal, però, tenir clar que Adobe Photoshop sempre convertirà i aplicarà el valor seleccionat en funció del mode de color de l'arxiu.

Exemple

Si seleccionem un color mitjançant el sistema HSB, però la imatge es troba en mode RGB, el color es recalculà i s'aplicarà en la imatge dins de l'espai de color (RGB).

Aquest aclariment resulta més evident encara si treballem amb una imatge en escala de grisos. I és que encara que el «Selector de Color» ens mostri un color (el que hàgim configurat, sigui quin sigui), aquest es reproduirà sempre en l'arxiu com una tonalitat acromàtica (gris). Aquest exemple remarca la diferència operativa entre el mode de color d'una imatge i el sistema de classificació de color que ofereix el programa.

2.8.1. Codificació HSB

La codificació HSB⁷ (*hue, saturation, brightness*) es basa en els tres paràmetres definidors del color. Per aquesta raó sovint esdevé un dels sistemes més intuïtius per a seleccionar colors i paletes cromàtiques des del disseny.

⁽⁷⁾To, saturació, brillantor.

El paràmetre to o matís⁸ constitueix en essència el color seleccionat. Determina la seva longitud d'ona dominant i per tant el valor que el defineix com a tal. Presenta els valors en graus (de 0 a 360), corresponents a la seva posició en la base o a la perifèria del con que s'utilitza per a representar l'espai cromàtic.

⁽⁸⁾Hue en anglès.

El paràmetre saturació⁹ correspondria a la intensitat del color seleccionat i es mesura en percentatge. Així, un 0% representa un acromatisme total o no saturació (blanc o negre) i progressivament es desplaça excèntricament des de l'eix central del con fins a un 100% que marca la total saturació, intensitat o puresa total del color.

⁽⁹⁾Saturation en anglès.

La brillantor¹⁰ determina la lluminositat del color i també es mesura en percentatge des del 0% (negre) en el vèrtex del con fins al 100% en la base, que correspon a la lluminositat màxima dels colors purs, incloent-hi el blanc resultant en el centre com a confluència de tots els components.

⁽¹⁰⁾Brightness en anglès.

Quan, des del disseny, cal configurar gammes de color harmòniques o determinats tipus de contrastos, pot resultar força útil recórrer al sistema HSB, ja que ens permet mantenir un paràmetre estable, per exemple, el to, i variar alhora els altres paràmetres de lluminositat o saturació.

Hi ha altres models que fan servir paràmetres molt similars a l'HSB, encara que en un ordre propi, de vegades amb valors diferents i/o representats tridimensionalment d'una altra manera. Alguns d'aquests són:

1) HLS¹¹ (*hue, lightness, saturation*).

⁽¹¹⁾To, luminància, saturació.

2) HSV¹² (*hue, saturation, value*).

⁽¹²⁾To, saturació, valor.

3) HVC¹³ (*hue, value, cromà*).

⁽¹³⁾To, saturació, color.

2.8.2. Codificació hexadecimal

El selector de color també ofereix la possibilitat de seleccionar o configurar el color mitjançant el sistema hexadecimal. Aquest és el sistema que utilitza el llenguatge HTML pel web.

L'escriptura hexadecimal utilitza setze dígit. A les deu xifres convencionals (del 0 al 9) s'afegeixen de manera successiva sis lletres de l'alfabet llatí (de la *a* a la *f*).

Taula d'equivalències decimal/hexadecimal entre zero i vint

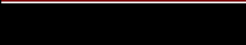








Decimal	Hexadecimal
0	0
1	1
2	2
3	3
4	4

Decimal	Hexadecimal
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
17	11
18	12
19	13
20	14



















Si per a representar el valor d'un component RGB amb el sistema decimal usem un rang entre 0 i 255, amb el sistema hexadecimal utilitzarem un rang entre 00 i FF, resultant el 00 el valor mínim i FF el màxim possible per a cada color primari.

Per a representar qualsevol color usarem tres parells de dígit, on cada parell determina el valor d'un dels tres components. Així, per a representar el negre (corresponent al valor 0, 0, 0 en RGB) en el sistema hexadecimal escriuríem 000000. Al contrari, el blanc es codificaria com a FFFFFFFF. Un vermell completament saturat seria FF0000, un verd completament saturat 00FF00 i un blau completament saturat 0000FF.

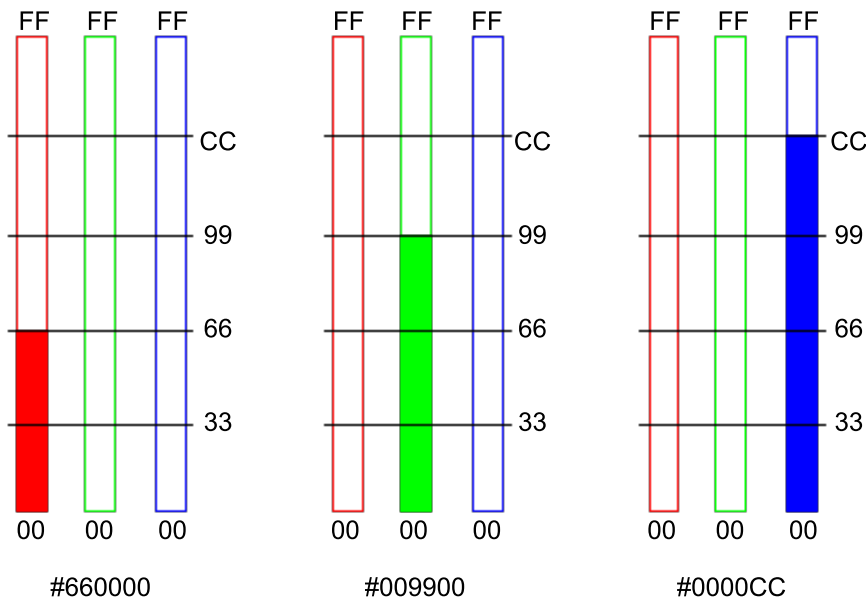
Colors RGB i notació hexadecimal

Color	Color RGB en hexadecimal	Color RGB decimal
	#000000	rgb(0, 0, 0)
	#FF0000	rgb(255, 0, 0)
	#00FF00	rgb(0, 255, 0)
	#0000FF	rgb(0, 0, 255)
	#FFFF00	rgb(255, 255, 0)
	#00FFFF	rgb(0, 255, 255)
	#FF00FF	rgb(255, 0, 255)
	#C0C0C0	rgb(192, 192, 192)
	#FFFFFF	rgb(255, 255, 255)

Variant els valors en cada component de l'RGB podem representar numèricament milers de colors. Utilitzant el sistema hexadecimal, com més baix és un valor (més pròxim a 00) menys llum tenim d'un component, i com més alt (més pròxim a FF) més en tenim. En disseny web els colors considerats *safe*, que es poden reproduir fidelment sense canvis en diferents navegadors i sistemes operatius, són els que contenen els nivells de 00, 33, 66, 99, CC o FF, per a cada color primari.

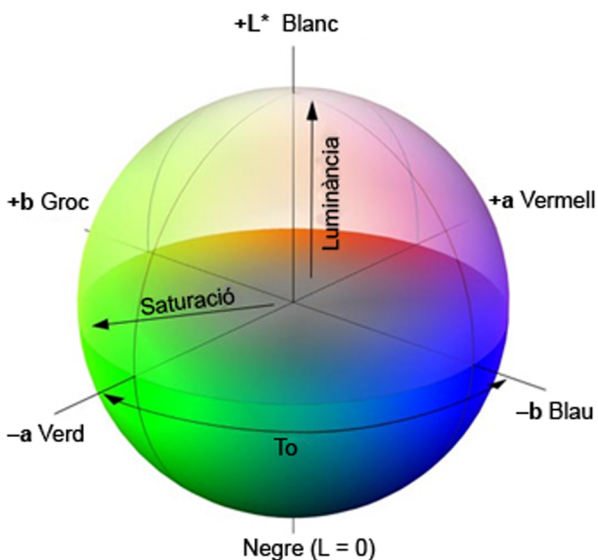
00	33	66	99	CC	FF	
						Component vermell
						Component verd
						Component blau

Un vermell fosc es pot representar com un #330000. Aquest vermell seria menys brillant que el vermell total (#FF0000). Un vermell amb més brillantor podria ser #660000. I faríem el mateix amb els altres components, per exemple, un verd total, #00FF00, o un #003300 verd fosc; un blau mig #000099 o un blau fosc #000033.



2.8.3. Codificació CIE L*a*b

El 1931 la CIE (Commission Internationale de l'Éclairage) va crear el sistema de codificació del color CIExyz. Aquest sistema seguia el canvi de paradigma científic del moment que passava a classificar el color en funció de la percepció subjectiva de l'espectador en lloc del model anterior geomètricament regular. El 1976 es va perfeccionar el sistema sota la denominació de CIE L*a*b.



© Josep Giribet. Creative Commons Reconeixement Compartir-Igual 3.0-es.

Els tres paràmetres que el determinen són:

1) **L = Lluminositat:** eix del negre (valor 0) al blanc (valor 100).

1) **a = Component cromàtic:** eix del verd (valor mínim -128) al vermell (valor màxim 127).

1) **b = Component cromàtic:** eix del blau (valor mínim -128) al groc (valor màxim 127).

L'ús de CIE L*a*b com a sistema de selecció de color podria semblar, en principi, menys intuïtiu que, per exemple, el sistema HSB.

2.8.4. Biblioteques de color. Les tintes planes o directes

La reproducció cromàtica per quadricromia (CMYK), i, per tant, per impressió pot resultar crítica. I és que no tots els colors RGB (mode de treball recomanable en autoedició) són imprimibles i en aquells que sí que ho són la consistència cromàtica sempre resulta delicada. No sempre els colors que el dissenyador visualitza en pantalla seran exactament els mateixos que final-

ment s'imprimiran sobre el suport (paper) final. Hi ha diverses respostes davant d'aquest fenomen. Una pot ser la utilització de les anomenades tintes planes o directes.

El seu nom les diferencia de les tintes procés CMYK per a la impressió convencional. Recordem que aquestes permeten la construcció cromàtica per mitjà de la síntesi sostractiva i partitiva. Les tintes planes, pel contrari, imprimiran directament el seu color. Per això els programes d'autoedició compten amb les anomenades biblioteques de color de cada fabricant per tal de carregar i mostrar en pantalla (de forma simulada) el color de la tinta final a imprimir. En aquest sentit, convé alhora que el dissenyador compti també amb una carta de color impresa del fabricant per a una selecció i identificació fiable.

Existeixen diferents catàlegs de tintes d'impressió útils per a seleccionar un color determinat. Per una disfuncional traducció literal de l'anglès, les *colour libraries* són anomenades biblioteques de color. Els fabricants són diversos. Cada indústria (arts gràfiques, tèxtil, automobilística, etc.) compta amb les seves. Però per la producció gràfica, i especialment aquella dirigida a la impressió convencional en òfset, esmentem una de les biblioteques de color més comunes com és Pantone. També n'hi ha d'altres com Trumatch, Focoltone, Toyo o Anpa-Color. La selecció d'un fabricant o un altre ha d'estar coordinada amb l'impressor i sovint es troba condicionada per l'àmbit geogràfic d'impressió.

L'ús de catàlegs de color per a gràfics digitals significa que aquests s'hauran d'imprimir o estampar usant aquestes tintes concretes. Com que les tintes estan codificades utilitzarem la mateixa codificació en l'autoedició digital.

Cal precisar que l'ús de cadascuna d'aquestes tintes suposa un cost important per a la producció. Per tant, convé valorar amb detall els avantatges que es guanyen (ampliació del rang cromàtic, impressió de colors corporatius...) envers la (possiblement més assequible) quadricromia. En tot cas, hauríem de contenir, tant com sigui possible, el nombre de tintes per garantir una impressió viable. En aquest sentit, és recomanable escollir tintes de catàleg quan vulguem imprimir menys de quatre colors o quan vulguem afegir un color molt específic impossible d'imprimir per quadricromia.

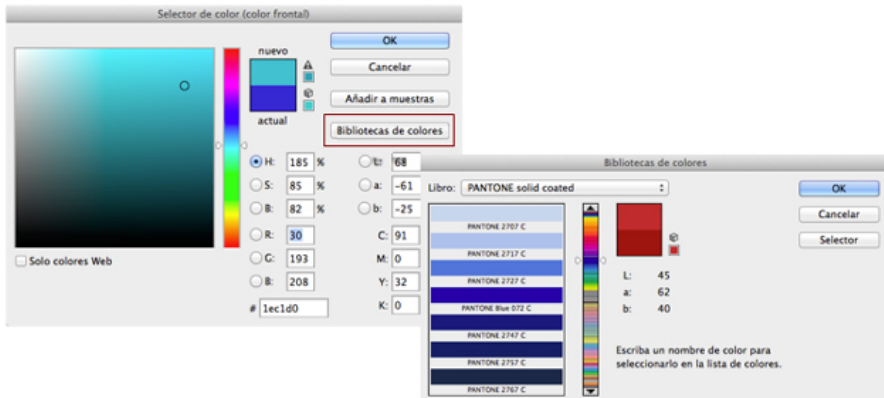
Un programa de mapa de bits com Adobe Photoshop permet treballar amb tintes planes o directes a dos nivells. Per una banda, podem estructurar la imatge segons aquestes a través d'un mode de color específic com és l'anomenat mode monotò, bitò, tritò o quadritò (accessible des del mode escala de grisos). Aquest mode permet treballar només amb una tinta plana o amb una combinació d'aquestes. D'altra banda, independentment del mode de color, podem també utilitzar aquest catàleg en el «Selector de Color» del programa. Seleccionarem així aquell color que desitgem per aplicar en la imatge. Recordem, però, que tot i la selecció de la tinta plana que en fem, el color s'aplicarà realment

Vegeu també

Per a més informació sobre la síntesi additiva, sostractiva i partitiva, podeu consultar l'apartat «3.3. Barreges: additiva, sostractiva, partitiva» del mòdul «Conceptes de disseny gràfic» dels materials de la nostra assignatura.

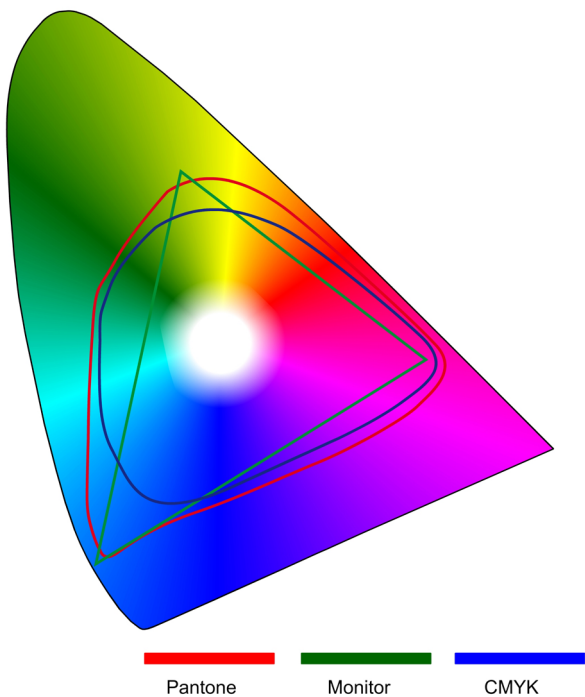
en funció del mode de color de la imatge. Per tant, si seleccionem una tinta d'aquesta biblioteca de color, però la imatge es troba en RGB, es recalcularan i aplicaran els colors corresponents.

Selector de Color d'Adobe Photoshop i biblioteca de colors corresponent a Pantone. Tria d'un color de la llibreria Pantone en el Selector de color d'Adobe Photoshop



Aquestes dues imatges es reproduïxen acollint-se al dret de citació o ressenya (art.32 LPI) i s'exclou de la llicència per defecte d'aquests materials.

Comparació visual dels rangs cromàtics que poden representar els sistemes assenyalats en el marc de l'espai CIE Yxy.



L'espai CIE Yxy s'aproximaria a l'espai cromàtic perceptible per l'ull humà.

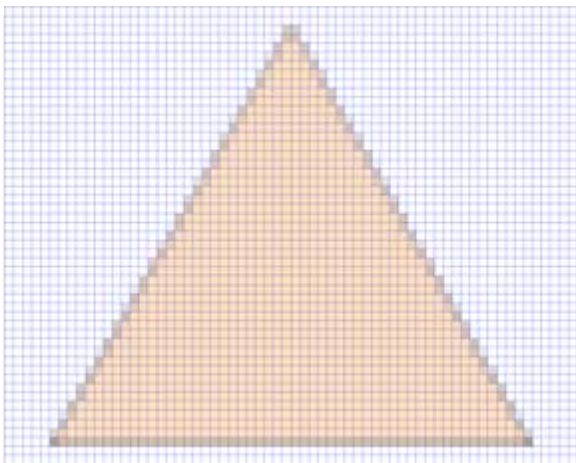
3. Gràfics vectorials

3.1. Estructuració en objectes vectorials

Els gràfics vectorials, a diferència de les imatges en mapa de bits, no es construeixen sobre la base d'una matriu de píxels equivalents, on cadascun pot reproduir un valor cromàtic diferenciat. No s'estructuren, doncs, podríem dir, punt per punt, sinó de manera més global. Aquests gràfics codifiquen la informació numèrica de la seva geometria. D'aquesta forma es generen objectes gràfics, independents entre si, definits per punts de coordenades, units tots ells per vectors.

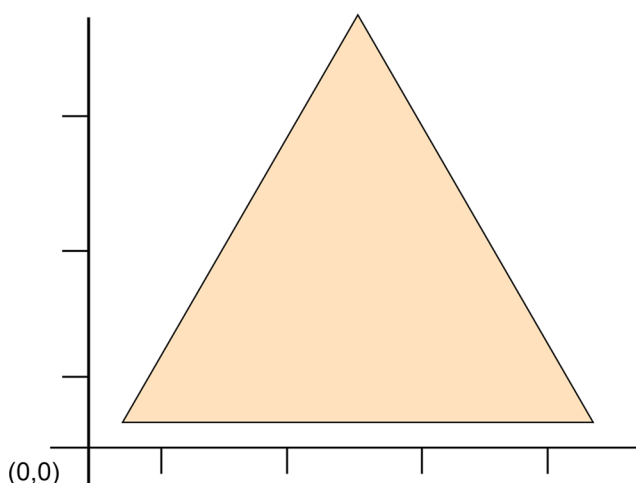
Aquests gràfics o objectes vectorials s'articulen en funció de fórmules matemàtiques que descriuen la posició, les dimensions, la forma i el color per tal de construir el que seran els dos elements constitutius de tot objecte vectorial, el contorneig i l'interior del grafisme. Aquesta arquitectura matemàtica permet el redimensionament il·limitat de l'objecte o conjunt d'objectes vectorials, sense afectar substancialment ni la càrrega de memòria ni la qualitat representativa.

Exemple d'un triangle en un gràfic de mapa de bits (a una resolució baixa)



Tenim informació del color de cada punt de la imatge (representats aquí per les caselles). No tenim informació del triangle per si mateix, aquest es mostra a partir de la informació dels punts. El gràfic està desat en un fitxer GIF.

Exemple d'un triangle en un gràfic vectorial



El triangle es defineix a partir de les coordenades dels vèrtexs i de la unió d'aquests a partir de línies rectes. Tenim, doncs, la informació geomètrica de l'objecte. El gràfic està desat en un fitxer SWF.

3.2. Escalabilitat

Cada objecte d'una imatge vectorial presenta unes característiques independents de la resta d'objectes de la imatge. Així, podem tenir objectes amb colors, mides i formes pròpies. La variació d'un o de tots aquests paràmetres només modificarà l'objecte seleccionat. Per tant, quan ampliem una imatge no redistribuïm els hipotètics píxels que la formen, ni modifiquem tampoc la resolució. L'únic que fem és variar els paràmetres de l'algorisme que calcula l'objecte.

Les imatges vectorials són caracteritzades amb el terme «escalable». Són escalables, perquè, tot i que ampliem la imatge o qualsevol dels objectes vectorials integrants, la qualitat visual no se'n ressentirà. La imatge no es pixelarà, ni tampoc, d'altra banda, augmentarà significativament el pes en memòria de l'arxiu.

L'escalabilitat dels objectes vectorials garanteix la reproducció (digital o impresa) amb la màxima definició possible. Això és possible perquè el llenguatge de descripció de pàgina (LDP) que transmet aquests gràfics es basa també en la mateixa codificació matemàtica (vectorial) i, per tant, en respecta la naturalesa fins a la reproducció pel dispositiu final. En aquest sentit cal tenir clar, doncs, que una imatge vectorial, sigui quina sigui la mida pertinent, es reproduirà o imprimirà sempre amb la màxima definició possible que permeti el dispositiu corresponent (evitant, per tant, el risc de pixelació al qual es troben exposades les imatges en mapa de bits).

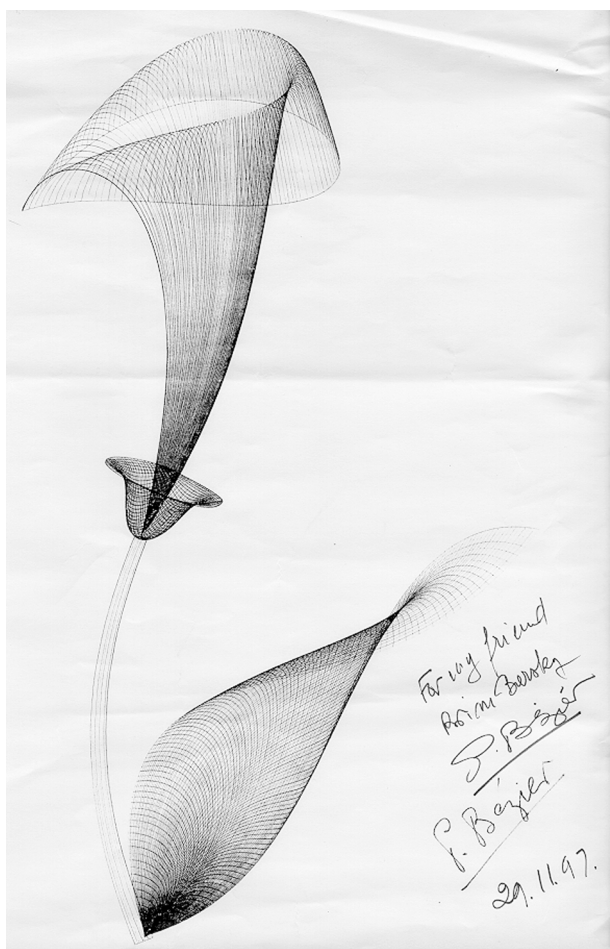
En conseqüència, l'objecte vectorial no depèn de la resolució i, en la majoria de casos, la mida d'emmagatzemament és molt inferior a la que tindria una imatge de mapa de bits.

3.3. Arxius vectorials i els objectes Bézier

3.3.1. Objectes Bézier

Hi ha diverses codificacions possibles d'una imatge a partir de vectors. La més estesa entre els programes i els formats de fitxer de gràfics vectorials és la basada en les anomenades corbes de Bézier o, per extensió, objectes Bézier.

Obra de Pierre Bézier feta utilitzant corbes de Bézier



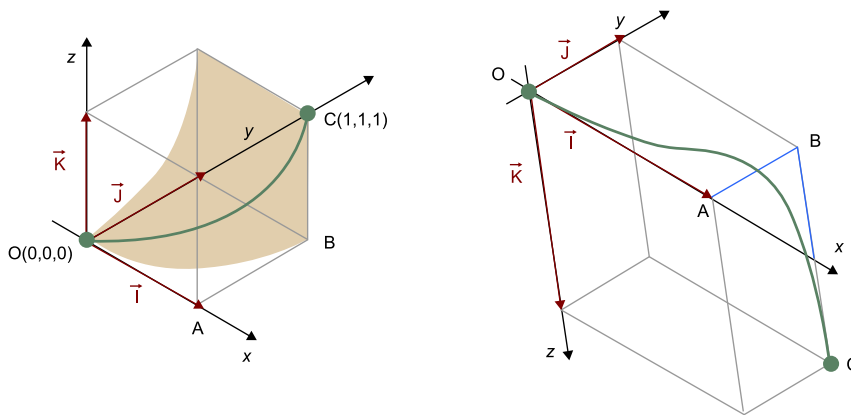
© Descendants Pierre Bézier. Aquesta imatge es reproduïx acollint-se al dret de citació o ressenya (art. 32 LPI), i està exclosa de la llicència per defecte d'aquests materials. Fotografia de Pierre Etienne Bézier.

3.3.2. Bézier en el programari gràfic

El programari de gràfics vectorials troba en les corbes de Bézier una metodologia tan fàcil d'usar com de computar. La manera de treballar és relativament similar a la ja explicada, tot i que amb alguns matisos.

Corbes de Bézier en gràfics 3D

A continuació podem observar l'ús de corbes de Bézier en programari de gràfics de dues dimensions. Tanmateix, Bézier va desenvolupar el seu sistema de corbes per a dissenyar carrosseries de cotxe. Per tant, pretenia corbar superfícies, no solament línies. El procediment serà el mateix. Per a entendre-ho podem començar descrivint una superfície corba dins d'un cub i després deformar aquest cub (canviant les posicions dels vèrtexs), per tal de modificar-ne així la corba inscrita.

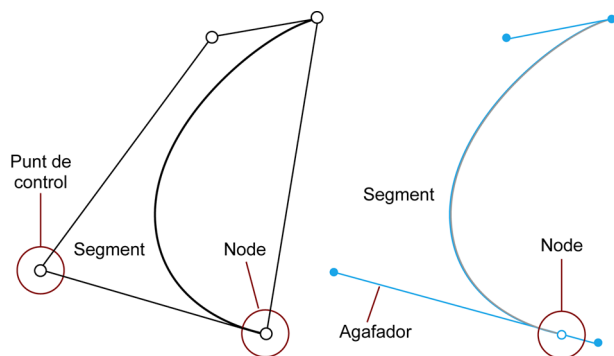


Un conjunt de corbes d'aquest tipus connectades entre si ens permetrien definir qualsevol superfície.

El programari de gràfics 3D (que escapa dels objectius d'aquest material) també fa servir les corbes de Bézier per a definir les seves formes.

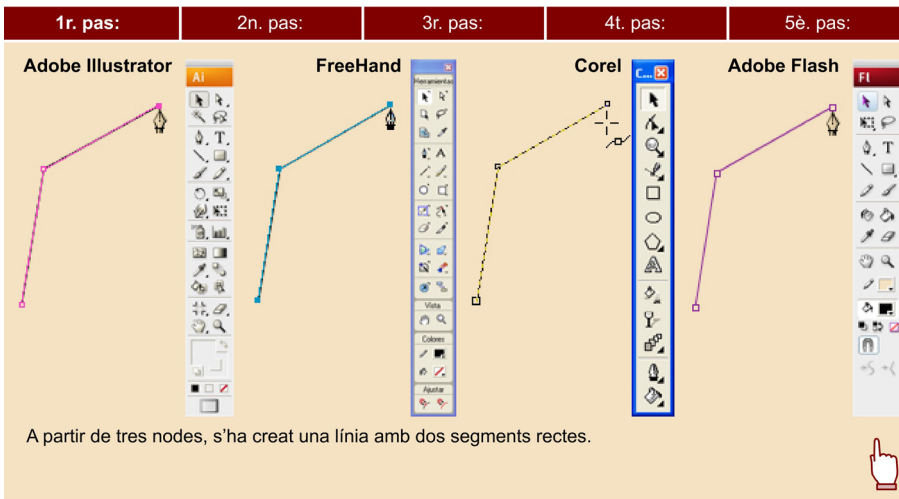
Ús de les corbes de Bézier en la geometria de les fonts tipogràfiques

En tipografia digital és necessari un sistema de codificació de la informació geomètrica que permeti que els tipus es visualitzin i imprimeixin perfectament a qualsevol mida. Per aquesta raó, el llenguatge de descripció de pàgina PostScript també recorre al sistema de Bézier. Aquesta sistema, doncs, és el que utilitzen les fonts TrueType (amb corbes de segon ordre) i PostScript Tipus 1 (amb corbes de tercer ordre).



Per a generar una línia es van creant els nodes, connectats entre si per línies rectes. Si es vol corbar un d'aquests segments, estirarem un agafador des de l'interior d'un node, que en realitat constitueix un punt de control o vèrtex del polígon. Regulant l'agafador corresponent de cada node es defineix la forma de la corba.

En els programes de gràfics l'eina ploma permet crear corbes de Bézier d'aquesta manera.



Podeu veure la seqüència de creació d'una corba amb programari vectorial. Mostrem captures de pantalla en quatre programes diferents (Adobe Illustrator, Freehand, Corel i Adobe Flash). Com es pot veure, la metodologia és molt similar i les diferències en la interfície són mínimes. Cliqueu en el botó per avançar o retrocedir en el procés de creació de la corba.

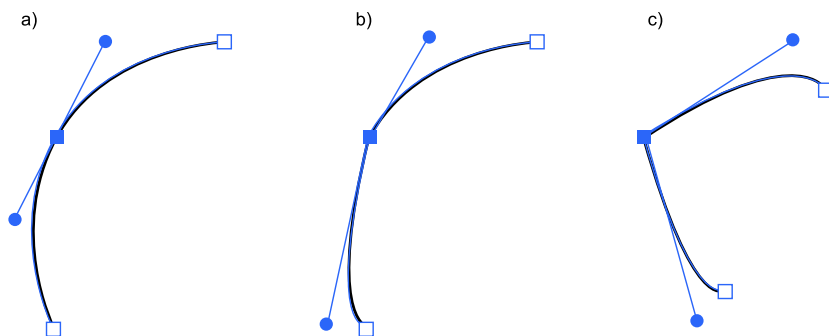
3.3.3. Treball de precisió amb les corbes de Bézier

Dibuixant objectes Bézier és fàcil crear elements amb massa nodes o amb corbes que no conserven continuïtat en tot el traçat. Convé, doncs, atendre una sèrie de normes que ens poden ajudar a treballar amb més precisió. En la taula següent presentem les més importants.

Normes	Incorrecte	Correcte
La línia 1/3: un agafador (distància entre node i punt de control) no ha de fer més d'1/3 del segment al qual modifica.		
Línia empremta: s'ha d'utilitzar el nombre de nodes més petit possible; per tant, en un arc no s'haurien d'utilitzar més de dos nodes.		
Línia tangent: l'agafador ha de ser tangent al segment, però mai creuar-lo.		

Encara que en formular-les com a «normes» usem un to categòric, no s'ha de perdre de vista que la pràctica gràfica també presenta un caràcter experimental. De fet, aquestes normes són el resultat de l'experiència pràctica i d'aquí n'emana el valor.

Un altre aspecte important que cal tenir en compte és la diferència entre punts de vèrtex i punts de corba. Quan connectem dos segments corbs, si volem que la corba tingui una bona continuïtat, les dues tangents del node que comparteixen els segments han d'estar alineades. En canvi, si el node fa la funció de vèrtex, les dues tangents no estaran alineades.



a) Línia definida per dos segments corbs correctament connectats entre si. Fixeu-vos que les tangents del node central estan alineades. El resultat és una corba que flueix sense «ruptures».
 b) Línia amb les tangents del node central no alineades, per la qual cosa la corba no presenta bona continuïtat.
 c) Gràfisme amb un node central que no actua com a vèrtex, per la qual cosa deliberadament els dos segments no formen una línia contínua. Les tangents no estan alineades i formen un angle.

Aquí el programari de gràfics ens facilita la tasca permetent que els dos agafadors es moguin conjuntament o siguin independents. La major part de programes de dibuix vectorial permeten que un punt sigui tant de tipus vèrtex com de tipus corba, podent passar d'un tipus a l'altre, en tot moment, i canviar les propietats del punt.

3.4. Propietats dels objectes vectorials

Les propietats principals d'un objecte vectorial són les següents:

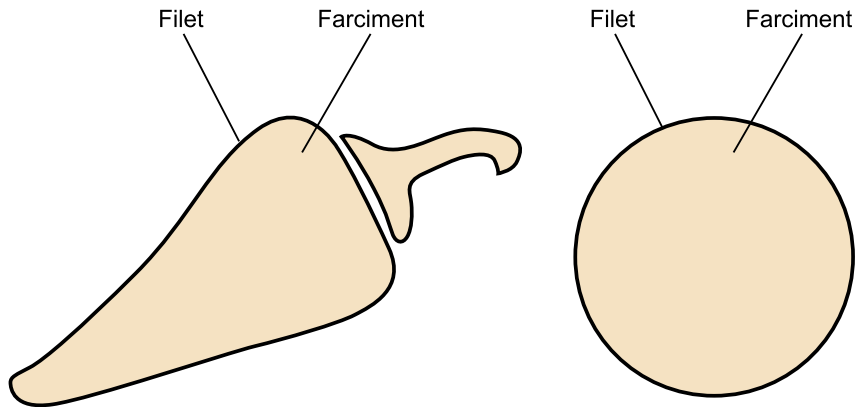
1) **Color interior o farciment.** Determina l'interior cromàtic de l'objecte. Aquest pot ser un color pla, una textura o un degradat. Per tal que un objecte es pugui omplir ha d'estar tancat (l'últim node s'uneix amb el primer).

2) **Filet, contorneig o línia.** Representa la línia externa que envolta l'objecte. Podem configurar el color, el tipus de traç (línies discontinües, punts, etc.) i el seu gruix.

3) **Posició.** Determina la situació de l'objecte en el document de treball. Els objectes vectorials es distribueixen en un sistema de coordenades de dues dimensions (x, y) , la qual cosa equival a un pla.

4) **Mida.** Determina l'àrea que ocupa l'objecte en el document de treball.



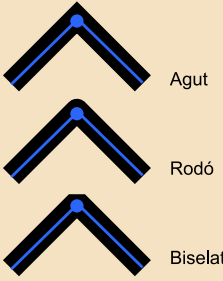
Propietats de l'objecte vectorial



3.4.1. Propietats de la línia o filet







Les propietats del filet poden variar d'un programa a l'altre. Bàsicament podem diferenciar les propietats que permeten els programes de «primera generació» destinats a impressió (Illustrator, Corel) i les que possibiliten els de «segona generació», pensats per a la reproducció final en pantalla (Fireworks, Flash).

Els programes, originàriament destinats a crear il·lustracions per a la impressió convencional, donen molta importància a la forma de la línia. S'hi poden controlar propietats com les que es mostren en la taula següent:

Propietats de la línia	
<p>6pt</p> <p>12pt</p> <p>24pt</p> 	<p>Grossària: es reparteix (a parts iguals) entre la part inferior i superior de la línia imaginària marcada pels dos nodes. L'amplada se sol mesurar per punts de pica (unitat de mesura usada pels tipògrafs). En el programari orientat a web, les mides són en píxels.</p>
	<p>Acabament: la línia pot acabar a la mateixa altura de l'últim node o bé pot tenir la mateixa grossària que l'amplada. També pot tenir un acabament rodó o, en alguns programes, en forma de punta, de fletxa o de cercle.</p>
 <p>Agut</p> <p>Rodó</p> <p>Biselat</p>	<p>Vèrtex: la forma dels vèrtexs (nodes no terminals de la línia) també es pot definir. Normalment, s'ofereixen tres opcions: agut, rodó i biselat.</p>

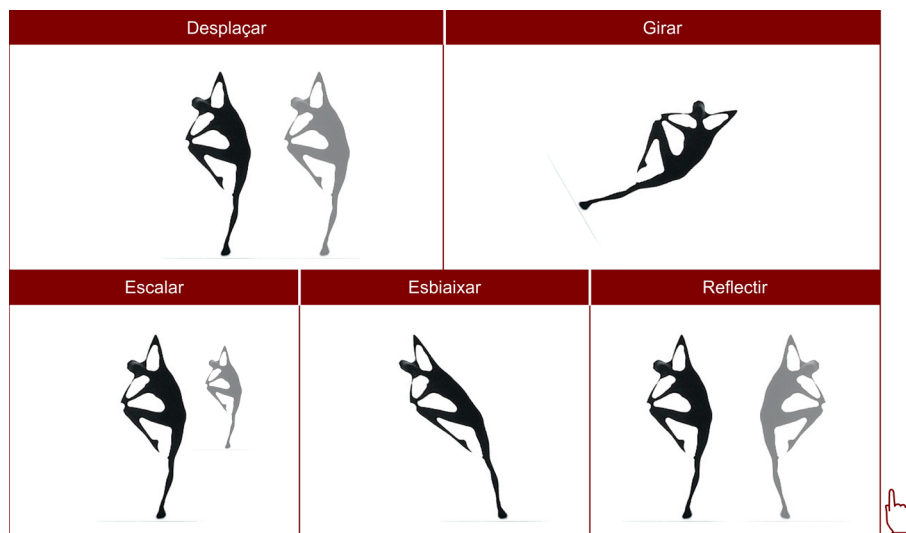
Els programes posteriors ideats per a crear gràfics vectorials, també per la seva publicació a la xarxa, insisteixen més en la visualització en pantalla. Per això introdueixen elements com el suavitzat (o *antialiasing*) per tal que la lí-

nia no aparegui dentada. També introdueixen la simulació de traços d'eina (cal·ligràfic, carbó, aerògraf, etc.) i textures que puguin donar caràcter a la línia, aproximant-se, de fet, a les possibilitats del programari de mapa de bits.

Exemple	Descripció
	<p>Línia bàsica "dura" (quan la línia és inclinada en veiem el dentat), de 10 píxels d'amplada, sense vora i sense textura.</p>
	<p>Línia bàsica "suau" (antialiàsing), amplada de 10 píxels, amb vora difuminada 50 i sense textura.</p>
	<p>Línia bàsica amb acabament arrodonit "suau" (antialiàsing), amplada de 30 píxels, difuminat de vora de 6 i sense textura.</p>
	<p>Línia bàsica amb acabament arrodonit "suau" (antialiàsing), amplada de 30 píxels, difuminat de vora de 90 i amb una textura ("Sorra") a un 40%.</p>
	<p>Línia bàsica amb acabament arrodonit "suau" (antialiàsing), amplada de 40 píxels, difuminat de vora de 90 i amb una textura ("Sorra") a un 95%.</p>
	<p>Línia "d'aerògraf", amplada de 30 píxels, difuminat de vora de 60, sense textura.</p>

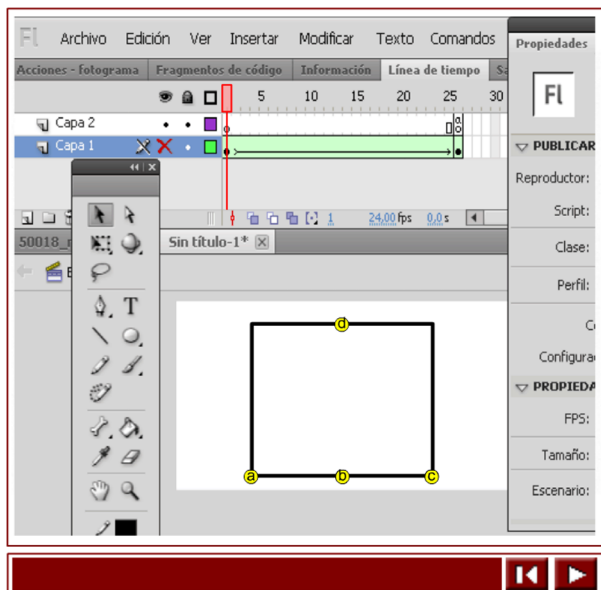
3.4.2. Transformacions dels objectes vectorials

Els programaris vectorials permeten qualsevol transformació geomètrica i, de fet, faciliten una eina o procediment específic per a l'aplicació efectiva sobre els objectes vectorials. Seguidament mostrem les transformacions més elementals.



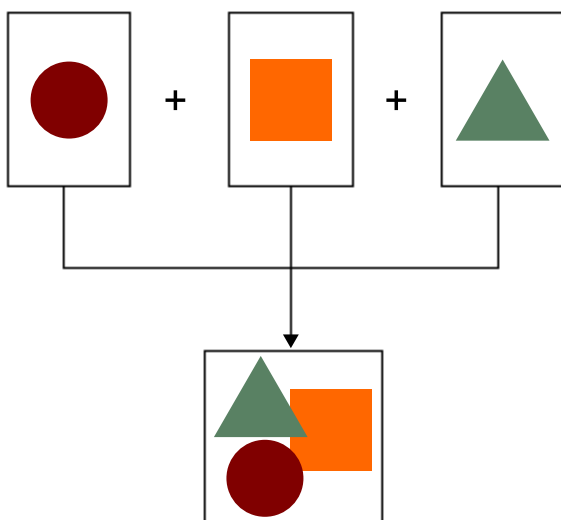
Totes aquestes transformacions es poden controlar numèricament amb precisió o bé realitzar-se de manera intuïtiva sobre el propi objecte.

Un altre recurs que permet el programari vectorial és la fusió d'un objecte amb un altre de diferent. D'aquesta manera es generen objectes intermedis que formen una seqüència de metamorfosi. Es pot escollir el nombre de passos, i en alguns programes (com Flash) és possible inserir una sèrie de punts (anomenats consells de forma) que permeten controlar on anirà a parar un punt de l'objecte inicial en l'objecte final.



3.4.3. Composició de gràfics vectorials

Composició en plans o capes dels objectes vectorials

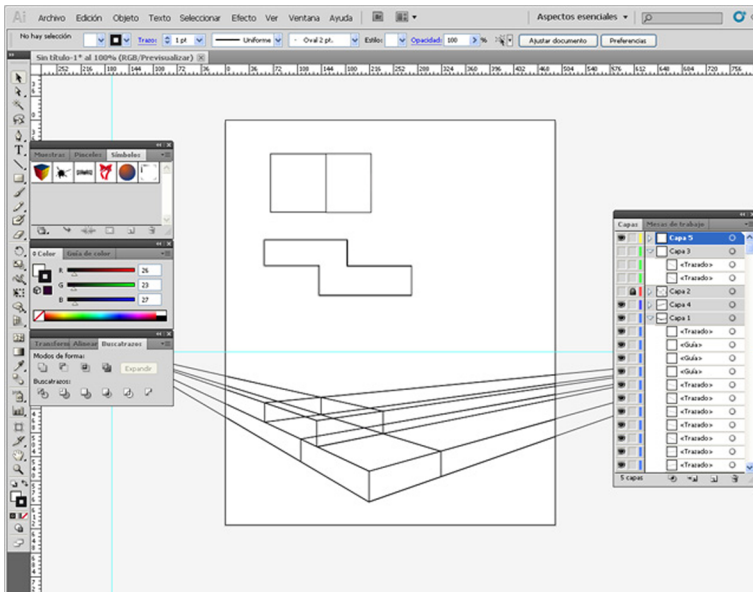


Les imatges vectorials s'estructuren sobre la base de diferents objectes independents. Aquesta estructuració autònoma dels objectes vectorials permet la manipulació de cadascun d'aquests per separat.

Tornant a l'exemple anterior, els objectes no se situen en el mateix pla geomètric. Es troben superposats i, de fet, els podem intercalar entre si. Aquesta operativa habitual per a la majoria de programes vectorials (Illustrator, Fireworks, CorelDraw, Inkscape, Xara Xtreme) no és possible, pel contrari, en el programari Flash. En aquest programa els objectes comparteixen un mateix pla geomètric, i en superposar-los se sumen automàticament (tret que estiguin agrupats).

A més, la majoria de programes ofereixen la possibilitat de treballar amb capes. Cada capa és independent i totes s'ordenen en una jerarquia de superposició que determina l'usuari. Treballar amb capes permet organitzar millor el treball. Les capes són independents i es poden bloquejar o ocultar. Cada capa, a més, ofereix diversos plans en funció del nombre d'objectes que contingui.

Treball amb capes a Fireworks



4. Formats d'arxiu per a gràfics

4.1. Tipus de format

La imatge digital es construeix, com hem indicat, segons dues arquitectures de codificació diferenciades:

1) **Mapa de bits:** s'estructura la imatge en una retícula de píxels, on cadascun d'aquests pot reproduir una tonalitat diferenciada.

2) **Vectorial:** es desa la informació geomètrica de cada objecte segons funcions matemàtiques.

Cada imatge, però, independentment de la seva naturalesa (*bitmap* o vectorial) pot ser codificada i desada en diferents formats d'arxiu. La raó d'aquesta diversitat podem relacionar-la amb l'afany de la indústria i de la investigació per donar resposta a les diferents necessitats de treball que van apareixent.

Convé, doncs, assimilar que cada format d'arxiu ofereix uns recursos diferenciats i que, per tant, caldrà optar per un format o un altre en funció de la finalitat de la imatge digital que s'ha de tractar.

Podem classificar els formats gràfics digitals segons diversos criteris.

1) **Arquitectura digital:** un arxiu pot estructurar-se en píxels (mapa de bits) o de forma vectorial. Cal afegir a aquesta dicotomia els anomenats metafitxers. Aquests formats admeten en únic arxiu gràfics vectorials i imatges en mapa de bits, tot respectant l'estructura de cadascun dels gràfics.

2) **Compressió:** una imatge digital, especialment si és de naturalesa *bitmap*, pot presentar un gran pes en memòria. Per aquesta raó els diferents formats incorporen mètodes de compressió que permeten una gestió i comunicació més eficient de l'arxiu. Els tipus de compressió són diversos i convé especialment distingir entre formats compressors amb pèrdua i sense pèrdua d'informació.

3) **Compatibilitat:** distingim entre formats nadius i formats compatibles. Aquesta diferenciació es fonamenta en funció de la naturalesa propietària o oberta de l'aplicació corresponent i, per tant, del format. També convindria tenir en compte la condició (real) multiplataforma o no.

4) **Finalitat:** en funció dels recursos que ofereixi o permeti un format, podria resultar més adient per un objectiu o un altre. En aquest sentit distingirem els formats en funció de la previsible sortida de la corresponent imatge digital. Diferenciarem, doncs, entre formats per pantalla i formats per impressió.

Per a una major síntesi i claredat abordarem, seguidament, els formats segons la primera distinció apuntada entre formats vectorials i formats en mapa de bits.

4.2. Formats per a gràfics vectorials i metafitxers

Alguns dels formats que anomenem vectorials funcionen en realitat com a metafitxers, ja que admeten gràfics de mapa de bits incrustats i conserven alhora la naturalesa vectorial del text i dels altres grafismes vectorials.

La capacitat de tot arxiu vectorial per gestionar els elements gràfics com a objectes independents és el que permet, en realitat, que els formats vectorials incorporin també elements (incrustats) en mapa de bits.

4.2.1. EPS, un primer format vectorial per a arts gràfiques

Durant molts anys el format EPS (*encapsulated postscript*) va ser un dels formats més habituals en les arts gràfiques. Progressivament, però, la seva utilització ha estat superada pel desenvolupament i extensió del format PDF.

EPS es basa en el llenguatge de descripció de pàgina PostScript¹⁴. Aquest llenguatge recorre a les corbes de Bézier cúbiques per a la descripció dels objectes. A més d'aquesta informació geomètrica, EPS codifica la posició dels grafismes dins d'un marc. Un marc en el qual també pot incrustar gràfics de mapa de bits, gràcies a la seva condició de metaformat. Cal indicar, però, que EPS és un format tancat. Una vegada desat, ja no permet l'edició del contingut, tot i l'estructuració vectorial. De fet, aquesta limitació explicaria la referència a l'encapsulació (*encapsulated*) de la denominació.

⁽¹⁴⁾PostScript és un llenguatge de descripció d'objectes vectorials desenvolupat durant la segona meitat dels setanta i la implementació final del qual va anar a càrrec de John Warnock, qui poc després fundaria Adobe Systems Inc.

D'altra banda, la naturalesa PostScript del format EPS en limita (o condiona) l'ús, ja que necessita d'impressores capacitades per interpretar aquest llenguatge independentment de la plataforma i del dispositiu. Altrament s'imprimirà la versió de previsualització associada a l'arxiu i no pròpiament aquest en les condicions òptimes.

4.2.2. PDF, un metaformat versàtil

Adobe va aconseguir amb el desenvolupament del format PDF (*portable document file*) superar significativament els recursos (limitats) que oferia, fins al moment, el mateix format EPS. I precisament un dels punts forts del nou format serà la versatilitat, ja que PDF esdevé igualment funcional tant per qualsevol tipus d'impressió com per l'edició i publicació per pantalla.

En aquest sentit, PDF facilita un gran avantatge per a la comunicació i l'edició posterior. Deslliura el receptor de l'arxiu de la necessitat de disposar de les tipografies utilitzades per a una edició i reproducció correcta del text. I és que PDF desa la geometria dels textos integrats i, si no troba en el sistema les fonts corresponents per a la reproducció i edició, les dibuixa a partir de les dades disponibles. Anteriorment, convenia «traçar» els textos per superar l'eventual mancança de les tipografies pel receptor de l'arxiu. Aquesta operativa, tanmateix, implicava que el text ja no podia ser editat posteriorment.

D'altra banda, la condició de metafitxer de PDF garanteix la integració en el document dels següents continguts, conservant l'estructura original:

- 1) suport de text, mantenint l'estructura vectorial (i per tant editable), i desant la informació corresponent que permet la seva gestió puntual, tot i no comptar amb les fonts originals de treball;
- 2) integració de gràfics vectorials, guardant l'estructura vectorial;
- 3) incrustació de gràfics *bitmap*, respectant les característiques pròpies (resolució...).

Però aquest format també incorpora, entre altres avantatges, recursos com ara:

- 1) condició multiplataforma real que garanteix que els elements de pàgina (gràfics i text) es mantinguin estables entre plataformes;
- 2) creació d'hipervincles i també enllaços interns amb continguts del document i altres recursos del sistema;
- 3) integració de característiques de navegació i interactivitat;
- 4) conservació i edició d'informació relativa a les metadades;
- 5) gestió i configuració de seguretat del mateix arxiu.

La primera versió del format PDF es va publicar el 1993. El juliol de 2008 el format fou designat per la International Organization for Standardization (ISO) com a estàndard (ISO 32000-1:2008). Es publica llavors la seva especificació

per tal que qualsevol desenvolupador pugui generar eines de suport. Adobe manté la propietat de diverses patents sobre el format, però en permet l'ús sense contrapartides mentre es compleixi amb aquesta especificació.

L'extensió actual del format PDF és innegable. Ha aconseguit consolidar-se com un format habitual de distribució electrònica de documents en gran part dels àmbits institucionals, professionals i particulars. La seva multicompatibilitat i relativa facilitat (accessibilitat, usabilitat...) per a la visualització justifiquen aquesta popularitat. També ha ajudat, sens dubte, la seva capacitat per a funcionar com a versió imprimible web. Amb la publicació de l'especificació, aquesta opció es va fer possible de manera automatitzada a partir de les dades en XHTML.

Com hem comentat, l'extensió del format no s'ha vist condicionada pel fet que PDF requereixi d'un visualitzador apropiat. Tot i la compatibilitat actual, compta amb una aplicació pròpia com és el paquet Acrobat d'Adobe. Utilitzem l'expressió paquet per indicar que Acrobat compta amb una sèrie de programes i utilitats amb funcions (i també accessibilitat) diferenciades. Així, per exemple, qualsevol usuari pot descarregar gratuïtament l'Acrobat Reader, com a extensió (*plug-in*) del navegador, per una visualització efectiva de qualsevol PDF. Podem també fer-ho mitjançant aplicatius lliures com EVince o KPDF. Per a una edició amb garanties de qualsevol arxiu PDF caldrà recórrer, però, a l'aplicació especialitzada i comercialitzable Acrobat Professional.

4.2.3. SWF, un format vectorial multimèdia

El format SWF (actualment reconegut com a acrònim d'*Small Web Format*) va ser un format desenvolupat per la companyia FutureWave Software, després adquirida per Macromedia, per a la creació i edició de gràfics vectorials amb animacions i interactivitat. També pot suportar vídeos, sempre que aquests s'incrustin en format FLV (Flash Vídeo).

Inicialment va ser el format natiu de l'aplicació Flash de Macromedia. Adobe, però, va absorbir aquesta companyia el 2005 i a partir de 2008 va iniciar un procés d'obertura de l'especificació perquè altres desenvolupadors poguessin generar eines relacionades i els cercadors d'Internet (com Google o Yahoo) poguessin indexar el contingut dels seus fitxers.

A diferència dels documents PDF, els arxius SWF sí que es poden incrustar dins d'una pàgina HTML, però requereixen disposar d'una extensió (*plug-in*) en el navegador. De fet, tant era així que Adobe amb cada nova versió del programa Flash publicava una versió actualitzada del format. Aquesta constant renovació obligava els usuaris web a una actualització permanent dels *plug-ins*.

Mentre els navegadors no tenien integrada la interpretació de fitxers SVG, el SWF es va convertir en un estàndard *de facto* per a fitxers vectorials i d'animació en l'entorn web.

4.2.4. SVG, un format vectorial obert per Internet

Durant molt de temps, i contràriament al que succeï amb els arxius de mapa de bits, un dels majors problemes dels formats vectorials va ser la seva incompatibilitat amb el llenguatge web HTML. Aquesta limitació va ser definitivament superada amb el desenvolupament del format SVG (*Scalable Vector Graphics*). Aquest format, basat en el llenguatge XML, compta amb el suport del W3 Consortium (W3C), l'organisme independent que defineix els estàndards de desenvolupament del World Wide Web. Actualment la majoria de navegadors suporten aquest format vectorial multicompatible.

SVG ofereix, a més de la condició vectorial, l'avantatge de suportar animacions i programació d'interactivitat a partir de llenguatges d'*script*. En ser un format obert, programes d'edició gràfica vectorial com Illustrator o Inkscape poden treballar i generar arxius en aquest format.

4.3. Formats per a mapa de bits

El ventall de formats per a gràfics de mapa de bits és ampli i divers. Un criteri diferenciador és la possibilitat de comprimir la informació integrada i, en cas de fer-ho, seleccionar el mètode de compressió efectiva. La qüestió de la compressió resulta fonamental ja que les imatges *bitmap* poden arribar a ocupar molta memòria. Una compressió eficient pot facilitar notablement la gestió, seguretat (evitar o reduir risc de corrupció), comunicació i impressió dels arxius.

4.3.1. TIFF, el format d'impressió

El format TIFF (*Tagged Image File Format*) va ser desenvolupat conjuntament per les companyies Microsoft i Aldus (aquesta última, posteriorment absorbida per Adobe).

La seva extensió i compatibilitat amb diferents plataformes, programaris i dispositius han fet d'aquest format pràcticament un estàndard en la producció gràfica actual per impressió. A aquesta multicompatibilitat cal sumar-hi la capacitat que ofereix el format per suportar múltiples recursos de l'edició gràfica *bitmap* com són la versatilitat de modes de color o el suport de capes i canals (recursos associats a programes com Adobe Photoshop o GIMP).

TIFF, per altra banda, incorpora la possibilitat de comprimir l'arxiu seleccionant el mètode de compressió. Aquest pot ser un mètode sense pèrdua d'informació com l'algoritme LZW (Lempel-Ziv-Welch). Es tracta d'un mètode de compressió i descompressió automàtica que allibera l'usuari de fer servir cap utilitat per descomprimir l'arxiu.

4.3.2. GIF, el format precursor per web

GIF (*Graphics Interchange Format*) va ser desenvolupat el 1987 per CompuServe per a l'intercanvi de gràfics en una llavors incipient tecnologia anomenada Internet. I precisament durant aquells primers temps de la xarxa va ser un dels formats de major utilització.

GIF codifica els gràfics només a 8 bits per píxel. Les imatges presenten, per tant, un pes fins a tres vegades menor que en JPEG o formats similars (que poden treballar a 24 bits per píxel). Ara bé, només permet una paleta cromàtica limitada a les 256 tonalitats que permet l'únic *byte* per píxel de memòria amb què treballa. No pot, doncs, reproduir el mode de color RGB. Aquesta capacitat cromàtica (limitada) és coneguda com a paleta indexada.

Paleta indexada

Aquest nom al·ludeix a l'assignació que en fa el format d'un número a cada color (fins a un màxim de 256) de la imatge. Correlativament s'associa el color corresponent a cada píxel per a reproduir, en definitiva, la imatge a color. Si una imatge presenta moltes varietats d'un color és possible incorporar una paleta on hi siguin totes a canvi de reduir la resta de colors. Per exemple, podríem definir una paleta amb 256 tonalitats de vermell, però la imatge només podria usar aquestes tonalitats i cap altra.

Cal contextualitzar, però, les limitacions del format en el moment en què va aparèixer. I és que les primeres xarxes de comunicacions demanaven especialment treballar amb arxius de pes molt reduït.

El 1989 CompuServe publicà una nova versió del format, anomenada GIF89a, que possibilitava designar un o diversos colors de la paleta com a transparents. Aquest tipus de recurs és conegut com a transparència d'índex i permet, de fet, integrar imatges siluetejades en el disseny web.

Actualment l'ús (reduït) del format GIF es limita a gràfics amb una paleta cromàtica mínima, com poden ser algunes imatges (preferentment no modulades) amb colors plans i possiblement siluetejades (com una icona o un imatgotip). Si recorreguéssim a GIF per desar imatges modulades a color (fotografies), la reproducció mostraria una disfuncionalitat apreciable com l'anomenada posterització o *banding* (agrupació dels colors de la imatge en bandes de color).



Una altra variant del format és el GIF animat que integra una seqüència de diferents fotogrames en GIF. La reproducció successiva d'aquests reproduceix una seqüència animada. Per a generar aquest tipus d'arxius cal recórrer a programes creats amb aquesta finalitat, o a d'altres que incorporen un mòdul d'animació GIF com Fireworks o Photoshop.

El format GIF utilitza el mètode LZW com a algoritme de compressió (sense pèrdua de qualitat). Excepte per a la variant animada, els arxius GIF es poden emmagatzemar en mode entrelaçat. Aquesta tècnica permet una reproducció progressiva de la imatge mentre es descarrega (en lloc de mostrar-se completa una vegada descarregada). Aquesta opció resultarà útil en xarxes amb un ample de banda crític.

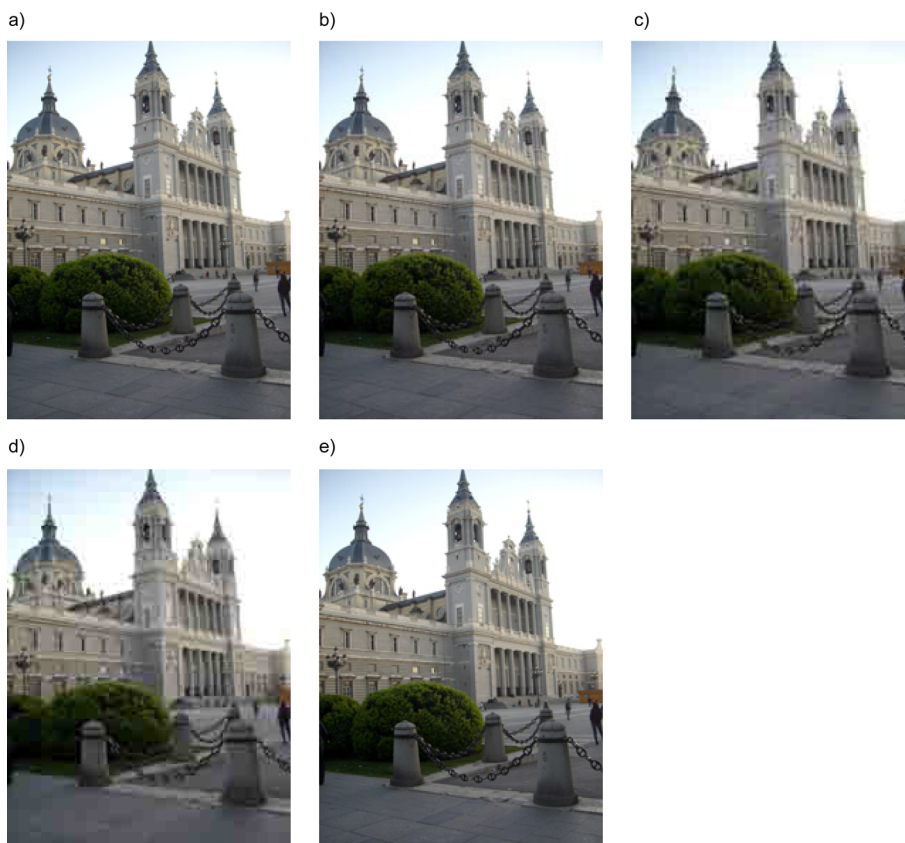
4.3.3. JPEG, més enllà de GIF

El format JPEG (*Joint Photographic Experts Group*) deu el seu nom al comitè creat per a consensuar un algoritme eficaç per a la compressió d'imatges fotogràfiques o de to continu, tant d'escala de grisos com en color. El format sorgeix, doncs, com a resposta a la necessitat de comunicar i reproduir per xarxa imatges a 24 bits per píxel i superar la limitada paleta cromàtica del format GIF.

Així doncs, JPEG suporta el mode RGB i, malgrat que també permet treballar només a 8 bits per píxel (imatges en escala de grisos), no admet el treball amb paletes indexades. En qualsevol cas, resulta adient per a imatges fotogràfiques. Reprodueix correctament degradats, però no tant contorns fins, detalls precisos o àrees de color pla. Per tant, no reproduïx amb consistència imatges amb línies o textos, tot i que disposem d'estratègies de compressió que poden minimitzar aquesta limitació.

Les dues limitacions més significatives del format JPEG són que no suporta cap tipus de transparència i la compressió amb pèrdua de informació.

El sistema de compressió del JPEG obté una ràtio de compressió superior a l'algoritme LZW o la que farà servir el format PNG. Constitueix un sistema molt eficient que aconsegueix reduir de manera significativa la mida dels arxius. Tanmateix, com dèiem, la compressió de JPEG implica pèrdua de dades. Aquesta pèrdua és regulable, però cal no perdre de vista que com més gran sigui la compressió més es perdrà la informació. De fet, una compressió total (i per tant molt incisiva) afectarà notablement a la reproducció de la imatge. El format permet configurar diferents qualitats relacionades amb el nivell de compressió. El rang més habitual especifica tretze nivells, del 0 al 12, essent 12 el nivell de major qualitat (i per tant de menor compressió) i a la inversa.



a) Compresió JPEG: 12. Mida: 68 Kb. b) Compresió JPEG: 6. Mida: 32 Kb. c) Compresió JPEG: 2. Mida: 25 Kb. d) Compresió JPEG: 0. Mida: 24 Kb. e) PNG-24 sense optimització. Mida: 4 Kb.

Les imatges JPEG admeten l'anomenada càrrega progressiva, tècnica semblant a la càrrega entrelaçada que utilitzen els formats GIF i PNG. Aquest recurs permet la reproducció en pantalla a baixa definició de la imatge simultàniament a la seva descàrrega. El sistema realitza escombratges successius que augmenten progressivament la definició de la imatge, fins a mostrar la versió definitiva.

Una imatge JPEG amb càrrega progressiva



JPEG 2000

L'any 2000 el comitè responsable de la versió original de 1992 va desenvolupar aquesta versió optimitzada del format JPEG.

La nova versió aporta uns nivells de compressió encara superiors, provant de corregir els defectes recurrents en els nivells alts de compressió de la versió anterior. La pràctica ha demostrat, però, que en algunes imatges el resultat pot presentar una nitidesa encara menor. En tot cas, la càrrega progressiva també es millora. Cal tenir en compte, però, que la seva visualització en alguns navegadors pot resultar crítica.

4.3.4. PNG, un format definitiu per a la xarxa?

Com a resposta a les restriccions en l'ús de l'algoritme de compressió LZW (situació que detallarem més endavant), la comunitat de programari lliure va emprendre el desenvolupament d'un nou format gràfic per al web. El resultat va culminar en el nou format PNG (*Portable Network Graphics*).

Existeixen tres versions en funció del nombre de bits usats per a la codificació del gràfic: PNG8, PNG24 i PNG32. PNG8 suposa la versió reduïda amb paleta indexada, mentre que PNG24 i PNG32 oferirien la versió desenvolupada que suporta el mode RGB. La primera versió, en tot cas, resultarà suficient per treballar amb imatges planes, mentre que convindrà recórrer a les versions PNG24 i PNG32 per a imatges modulades a color (fotogràfiques).

La versió PNG8 admet transparència d'índex. No així la versió PNG 24, tot i que la versió PNG32 pot guardar canal alfa i per tant admet diferents nivells de transparència. Aquest recurs, per exemple, permet guardar el siluetejat d'una figura amb els contorns suavitzats respecte del fons.

PNG compta amb un sistema de compressió lliure i sense pèrdua de qualitat conegut com a *deflate/inflate*, derivat de l'algoritme LZ77 (comprès a la mateixa «família» que l'LZW). D'altra banda, PNG també treballa amb la tècnica de càrrega entrellaçada, basada en el mètode Adam7 (més ràpid que l'utilitzat pel format GIF).

4.3.5. Transparència d'índex i transparència alfa

Hem comentat fins ara que els formats GIF i PNG suporten transparències, a diferència del format JPEG.

Cal diferenciar, però, els dos tipus de transparència amb què treballen aquests formats:

1) **Transparència d'índex:** un o més colors de la paleta són designats (i referenciats) com a transparents. No permet, de fet, semitransparències.

2) **Transparència alfa:** integració d'un canal alfa addicional que permet una aplicació independent i modulada de diferents valors de transparència (o opacitat) a la imatge.

Les versions GIF89a, PGN-8 i PNG-24 permeten aplicar transparència d'índex. En decidir quin color serà transparent cal controlar que la imatge no presenti píxels del mateix color en la figura, ja que aquests píxels també es mostrarien transparents. Una altra limitació d'aquesta transparència és que el contorn de la figura no pot reproduir-se suavitzat, ja que no permet diferents valors de transparència. La figura apareixerà, per tant, «retallada».



La transparència alfa, pel contrari, permet definir diversos valors de semitransparència. En no adjudicar aquests valors a colors concrets d'una paleta, les semitransparències es respectaran i reproduiran independentment dels colors que es trobin a la imatge. Per tant, podem siluetejar una figura amb els contorns progressius i evitar així la sensació de retall en la imatge. PNG-32 suporta correctament aquest tipus de transparència.

a)



b)



a) Transparència d'índex en una imatge GIF89a.

b) Transparència alfa en una imatge PNG-32. Aquest exemple fa servir el format PNG amb semitransparència per canal alfa.

4.3.6. Sistemes de compressió

Els sistemes de compressió redueixen la memòria que ocupa un arxiu. Aquesta compressió pot convenir per tal d'ocupar menys espai de disc, però resulta especialment recomanable quan l'arxiu s'ha de transferir i/o publicar a la xarxa. Com més reduït sigui l'arxiu, més ràpida serà la transferència de les dades.

Els sistemes de compressió eliminen redundàncies i informacions no significatives alhora que «resumeixen» la informació que s'ha d'emmagatzemar. Aquesta reestructuració de les dades es pot fer de dues maneres:

1) **Amb pèrdua de dades.** L'algoritme de compressió assimila dades relativament semblants en un únic valor (per exemple, diversos tons de blau es redueixen a un únic blau). En aquest resum de la informació es perden, per tant, detalls. Una informació suprimida que, en cap cas, es recuperarà amb la descompressió. Aquest tipus de compressió és el que utilitza, per exemple, el format JPEG.

2) **Sense pèrdua de dades.** Es busquen maneres d'optimitzar el registre de les dades eliminant la informació redundant i simplificant la manera d'expressar-la. Tota la informació «reduïda» es torna a recuperar en la descompressió. Aquest tipus de compressió és el que utilitzen, per exemple, formats com TIFF, GIF o PNG.

Cal tenir en compte, però, que tota compressió implicarà necessàriament una descompressió. I aquesta descompressió també necessitarà temps. En general, però, els sistemes de compressió/descompressió i les capacitats de processament dels dispositius actuals fan que el temps invertit en la descompressió resulti comparativament molt inferior al guanyat per la transmissió i/o gestió de l'arxiu comprimit.

JPEG, compressió amb pèrdua de dades

El sistema utilitzat per JPEG és molt eficient comprimint imatges modulades (fotogràfiques), però no tant, com ja hem comentat, amb imatges de colors plans o que continguin línies fines molt contrastades.

L'algoritme de compressió de JPEG, a grans trets, guarda la imatge separant la informació de brillantor de la dels matisos de color i elimina aquelles subtils diferències de color que considera no perceptibles. És un format, doncs, de compressió que elimina informació. El nivell de compressió és configurable, però cal tenir clar que per mínima que sigui la compressió, sempre es perdrà informació.

Compressió: 10



Compressió: 5



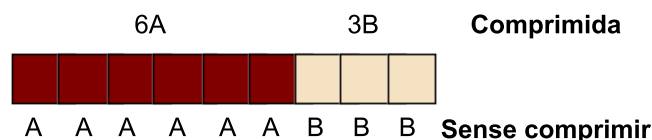
Compressió: 0



LZW i algoritmes relacionats, la compressió sense pèrdua de dades

LZW (Lempel-Ziv-Welch) és l'algoritme que usa el format GIF i, opcionalment, també el format TIFF per a la compressió dels gràfics. Es tracta d'un sistema de compressió sense pèrdua de dades. A grans trets, podríem dir que aquest algoritme busca i codifica aquelles seqüències consecutives de valors de color. Així redueix al màxim la redundància en la repetició de seqüència i, d'altra banda, permet que la descompressió recuperi, en decodificar, exactament els mateixos píxels de la imatge original.

RLE



Un sistema de compressió sense pèrdua de dades és el *run-length encoding* (RLE). Seqüències de dades consecutives amb el mateix valor s'emmagatzemen de forma «resumida» com un únic valor al qual li sumem el seu recompte.

Podeu observar de forma esquemàtica com resumiria un sistema de compressió RLE aquesta fila de 9 píxels per tal de resumir la informació.

RLE realitza un procés de reconeixement de seqüències consecutives de color, és a dir, recorre un per un el total dels píxels. Si troba més d'un píxel consecutiu del mateix color,

emmagatzema les dades d'aquest píxel i la corresponent quantitat en la seqüència. Així, per exemple, el resultat sintetitzaria una informació que presentés nou dígit (AAAAA-ABBB) en una expressió de només quatre dígit (6A3B).

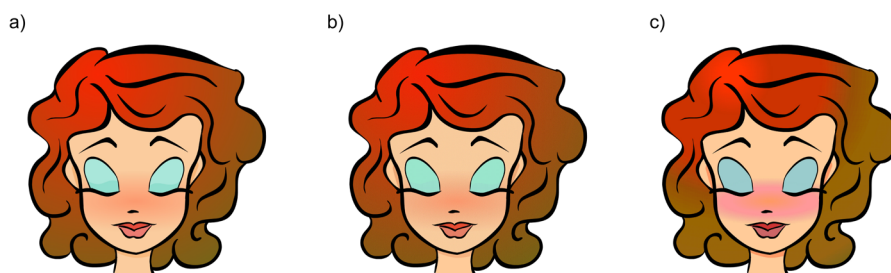
Els sistemes derivats d'LZ77 (com LZW) parteixen d'aquesta compressió com a base i la milloren tenint en compte la repetició en seqüències de dígit (com podria ser en AABAABAAB).

El sistema de compressió *deflate/inflate* que utilitza PNG es deriva també de l'algoritme LZ77 (utilitzat per zip, gzip, pkzip, etc.). Per tant, GIF i PNG utilitzen un mètode de compressió similar. El sistema *deflate/inflate* seria similar al descrit en la il·lustració de l'apartat anterior, però de manera optimitzada. Introdueix diversos filtres que preprocessen la informació de color per tal d'aconseguir una codificació encara més eficient. També és, per tant, un sistema de compressió sense pèrdues d'informació.

4.3.7. Recursos d'optimització pels formats GIF i PNG

Tot i que els algoritmes de compressió que usen els formats GIF i PNG no impliquen pèrdua de dades, aquesta pèrdua sí que es generarà en cas de desar la imatge en mode indexat. Cal recordar que aquest és l'únic mode disponible en GIF i que, de fet, també ho serà en desar la imatge com a PNG-8. Així doncs, qualsevol imatge RGB (i per tant amb més de 256 colors) patirà una pèrdua d'informació en desar-la en format GIF o PNG-8.

Alguns programes (com Fireworks o GIMP) permeten controlar el nivell de pèrdua d'informació per reduir la memòria de la imatge finalment exportada. Aquesta pèrdua es defineix per percentatge. El nivell recomanable variarà entre el 5% i el 15%.



a) Arxiu GIF original. b) Valor de pèrdua de 30. c) Valor de pèrdua de 100.

Tanmateix, existeixen altres recursos per a optimitzar la imatge que ofereixen encara més control.

Una alternativa consisteix a escollir una **paleta** determinada. Els programes de gràfics faciliten aquesta tasca amb paletes predefinides i opcions de parametrització com les següents:

1) **Paleta exacta.** Només viable per a imatges amb un màxim de 256 colors. El programa crea una paleta que contingui tots els colors de la imatge. La imatge no resulta, doncs, afectada.

2) **Paleta adaptable.** El programa rastreja els 256 colors més freqüents de la imatge i n'estableix la paleta corresponent. També es poden crear paletes adaptables a una quantitat inferior de colors.

3) **Paleta perceptual.** El programa genera una paleta de 256 colors adaptable, en què tenen prioritat els colors als quals l'ull humà és més sensible.

4) **Paleta predefinida.** Ajusta la imatge a una de les paletes disponibles amb el programa, com ara:

a) **paleta Windows** (paleta de colors del sistema operatiu Windows);

b) **paleta Macintosh** (paleta de colors del sistema operatiu MacOS);

c) **uniforme** (paleta creada a partir d'un mostreig uniforme dels colors de cada component RGB). Representa els colors originals de la imatge mitjançant el color més pròxim a la paleta escollida.

5) **Web216.** Paleta amb els 216 colors que els navegadors web utilitzen per a mostrar imatges en monitors limitats a 8 bits (cada vegada més escassos). També es coneix com a *websafe* o *browser-safe*. Si s'escull aquesta paleta, el programa adapta els colors de la imatge als colors més pròxims de la paleta.

6) **Adaptable WebSnap.** El programa adapta els colors de la imatge pròxims a la paleta Web216 i respecta els que se n'allunyen.

7) **Selectiva.** El programa crea una paleta semblant a la paleta perceptual però respectant els colors de la paleta Web216.

Quan s'utilitza una paleta predefinida, però, es corre el risc que contingui colors no integrats en la imatge original. Alguns programes disposen de la possibilitat d'eliminar els colors no utilitzats per tal que la imatge final ocupi menys espai de memòria.

En l'animació adjunta es pot comparar l'aplicació a una mateixa imatge de la paleta Web216 i d'una paleta Adaptable WebSnap amb diferent nombre de colors.



Imatge original
24 bits, pot representar 16.777.216 colors
Format PNG-24
Memòria: 163 k



Paleta Web216
216 colors comuns a sistema MacOS i Windows
Format PNG-8
Memòria: 29 k



Paleta WebSnap
8 bits, 256 colors. Tria un color de la paleta Web216 si troba un color semblant
Format PNG-8
Memòria: 51 k



Paleta WebSnap 128 colors
6 bits, 128 colors. Tria un color de la paleta Web216 si troba un color semblant
Format PNG-8
Memòria: 44 k



Paleta WebSnap 32 colors
4 bits, 32 colors. Tria un color de la paleta Web216 si troba un color semblant
Format PNG-8
Memòria: 30 k



Paleta WebSnap 8 colors
3 bits, 8 colors. Tria un color de la paleta Web216 si troba un color semblant
Format PNG-8
Memòria: 18 k



Paleta WebSnap 8 colors
3 bits, 8 colors. Tria un color de la paleta Web216 si troba un color semblant
Format PNG-8
Memòria: 18 k

Les imatges d'aquest exemple estan en format PNG.

Una altra manera de preservar certa fidelitat amb la imatge original en una imatge indexada és aplicar algun tipus de tramatge que permeti obtenir el color per mitjà d'una barreja partitiva. Així, els programes de gràfics ofereixen diverses possibilitats de tramatge.



1) **Motiu.** Fa servir un motiu quadrat regular que genera un efecte similar a la trama de semitons d'impremta per a una representació aproximada d'aquells colors que no es trobin a la paleta.

2) **Difusió.** Utilitza un mètode aleatori de difusió de punts de trama per a obtenir un tramatge menys estructurat que l'anterior. Generalment ofereix una relativa millor qualitat.

Entre les opcions del tramatge de difusió pot trobar-se disponible la configuració del percentatge d'aplicació. Aquest valor determinarà la substitució d'un color crític per un altre o, en el seu defecte, la representació mitjançant tram. Amb un tant per cent reduït, només es tramaran aquells colors sense un equivalent pròxim a la paleta, i a mesura que incrementem el valor el tramatge s'estendrà a totes les àrees de la imatge. Alguns programes també ofereixen la possibilitat de «protegir» alguns colors de la paleta, de manera que els colors coincidents entre la paleta i la imatge no quedin tramats.

En la imatge següent es mostra la modificació que experimenta una indexació en introduir el tramatge. Normalment aquest recurs permet representar més bé els degradats i clarobscur. Fixeu-vos com el pes de la imatge augmenta lleugerament amb el tramatge, ja que es generen menys píxels contigus del mateix color i la compressió resulta, per tant, menys eficient.

Exemples de tramatge

Paleta Web216 Format PNG-8	
	
La imatge no està tramada. Memòria: 29 k	La imatge té un tram de 100%. Memòria: 37 k
Paleta WebSnap Format PNG-8	
	
La imatge no està tramada. Memòria: 51 k	La imatge té un tram de 100%. Memòria: 54 k

D'altra banda, recordem finalment com els formats GIF i PNG suporten l'opció d'entrellaçament. Aquest recurs possibilita una reproducció immediata en pantalla a baixa resolució que progressivament anirà guanyant en qualitat a mesura que les dades arribin al dispositiu reproductor.

Aquest recurs n'augmenta lleugerament el pes, però permet que l'usuari visualitzi una versió aproximada de la imatge, tot just iniciar la descàrrega (de forma més ràpida amb el format PNG). Altrament una imatge sense entrellaçament es carregarà en sentit descendent i directament amb la qualitat final, però només una vegada completada la transmissió.

Dues imatges amb entrellaçament, una en format GIF i una altra en PNG



4.3.8. LZW i el desenvolupament del format PNG

Hem esmentat anteriorment com les restriccions d'ús de l'algoritme LZW van impulsar el desenvolupament del format PNG.

El sistema de compressió LZW (*Lempel-Ziv-Welch*) va ser desenvolupat l'any 1983 per Terry Welch a partir dels algoritmes LZ77 i LZ78, desenvolupats pels investigadors israelites Jacob Ziv al 1977 i Abraham Lempel l'any 1978, respectivament. El sistema es patenta als Estats Units l'any 1985, i actualment pertany a l'empresa Unisys.

Per la seva banda, Comuserve crea, l'any 1987, el format per a intercanvi de gràfics a Internet GIF (*Graphics Interchange Format*), que incorpora el sistema de compressió LZW, igual que ho farà el seu successor GIF89a. Durant els anys següents, i abans de l'arribada de JPEG, GIF es convertirà en l'estàndard tant d'intercanvi d'imatges a Internet com d'integració d'imatges en pàgines web. A finals de 1994, però, Unisys, que fins a aquell moment no havia emprès cap acció, reforçada per un acord amb Comuserve, comença a demanar el pagament de *regalies* o *royalties* als desenvolupadors de programari que suporten GIF.

Aquesta situació resulta incompatible amb els programes de codi obert. Alguns operadors de la comunitat de programari lliure es posen llavors en marxa per crear un format alternatiu. I ja durant el 1995 el desenvolupament avança ràpidament amb múltiples aportacions. S'utilitza l'algoritme LZ77, que sí que estava lliure de patents, i la codificació de Huffman. El 1996 finalment W3C anuncia la disponibilitat d'un nou format, anomenat PNG, del qual en recomana l'ús, atesa la condició lliure de patents. Comuserve comunica aleshores que la pròxima versió del format GIF (GIF24) abandonarà el mètode LZW i es basarà en l'especificació PNG. El juny de 2003 la patent de l'LZW expira als Estats Units, però Unisys adverteix que continua vigent en d'altres països. Tanmateix, segons The Software Freedom Law Center, a partir d'octubre de 2006 s'extingeixen les darreres patents rellevants sobre l'algoritme LZW.

Actualment, per tant, disposem d'un nou format com és PNG, que aporta nous recursos i es troba lliure de restriccions per al seu desenvolupament i utilització. Tot i que inicialment PNG va topar amb limitacions en la reproducció d'aquest format en alguns navegadors, el problema ha estat finalment superat per les noves versions.

Allò que resulta, en tot cas, interessant de concloure és que, tot i que no sempre els formats es troben lliures de restriccions, algunes vegades aquesta restricció pot tenir relació, com en aquest cas, amb el sistema de compressió utilitzat. En general, però, els propietaris dels formats no en restringeixen o condicionen l'ús, tot i que és cert que en alguns casos no permeten que altres persones o organitzacions desenvolupin la tecnologia del format.

4.4. Programari i formats nadius

A més dels formats analitzats, existeixen altres formats gràfics. Entre tots els possibles, tot seguit abordem aquells que són propis dels programes d'edició gràfica. Són els anomenats formats nadius o formats font del programa.

Els programes d'edició i tractament de gràfics necessiten un format apropiat per a editar i desar la informació durant la sessió de treball. Generalment es tracta d'informació addicional pròpia del programa que permetrà seguir amb l'edició en futures sessions de treball. Aquesta informació no ha de resultar necessàriament visible en la reproducció final de la imatge (com és el cas, per exemple, de la informació de capes).

Aquests programes permeten també desar o exportar en formats diferents als propis. Així, podríem tancar un gràfic en format PNG o JPEG per a la publicació web mentre conservem una versió en el format natiu del nostre editor de gràfics. D'aquesta forma podríem salvaguardar, per exemple, l'estructura de capes o la capacitat editable del text per a una possible modificació posterior.

Alguns d'aquests formats, puntualment, poden acabar convertint-se en estàndards *de facto* per diversos motius. Així, per exemple, una millora sensible en la forma de desar la informació, o la introducció de característiques noves, o simplement perquè el programa es converteix en un líder en el sector, poden convertir el format pràcticament en una convenció de treball. És el cas, no com a format de publicació però sí com a format d'intercanvi entre programes, del format natiu d'Adobe Photoshop (que presenta l'extensió «psd»). Així, altres programes com ara GIMP o Fireworks poden treballar amb arxius d'aquest format, tot conservant l'estructura de capes i altres dades de treball emmagatzemades. Un altre cas paradigmàtic és el del format PDF que, tot i ser un format propi del programari Adobe Acrobat, ha estat reconegut com un estàndard oficial internacional.

Pel que fa al programari lliure, una aplicació com GIMP ha aconseguit que el seu format natiu (XCF, *eXperimental Computing Facility*), s'hagi difós com a format d'exportació o importació per a altres programes d'edició gràfica també lliures.

Un cas ben diferent és que un nou programari assimili com a nadiu un format prèviament existent i a partir d'aquí el desenvolupi. Aquest va ser el cas del programa Fireworks amb el format PNG. La casa matriu d'aleshores, Macromedia, va incorporar en aquest format lliure característiques pròpies com ara la conservació de l'estructura vectorial dels gràfics corresponents i, per tant, la posterior recuperació i edició tant des del mateix programa com des de Flash, també llavors de Macromedia. Cal, però, aclarir com aquest format natiu (Fireworks) segueix utilitzant la mateixa extensió que el corresponent format lliure i genèric PNG. Aquest és un cas, i no l'únic, en què l'extensió d'un arxiu no ens informa prou bé sobre el contingut d'un fitxer.

4.5. Taula informativa dels diferents formats

En la taula següent recollim breument les característiques bàsiques d'alguns dels formats més habituals.

Formats <i>bitmap</i>	Característiques
JPEG <i>joint picture expert group</i> *.jpeg *.jpg.	Format mapa de bits. Mode RGB (24 bits). No admet transparències. Mètode propi de compressió amb pèrdua de dades. Admet càrrega progressiva. Adequat per a imatges modulades com les fotogràfiques per al web.
GIF <i>graphical interchange format</i> *.gif. (versions: GIF87, GIF89a i GIF animat)	Format mapa de bits. Mode indexat (8 bits). Admet transparència d'índex (GIF89a). Mètode de compressió LZW sense pèrdua de dades. Admet càrrega entrelaçada. Adequat per a imatges amb colors plans (web). Desenvolupat i propietat de CompuServe. Mètode de compressió LZW patentat per Unisys.
PNG <i>portable network graphics</i> *.png. (versions: PNG-8, PNG-24 i PNG-32)	Format mapa de bits. Mode indexat (PNG-8) o RGB (PNG-24 i PNG-32). Admet transparència d'índex (PNG-8) i transparència alfa (PNG-32). Mètode de compressió <i>deflate/inflate</i> sense pèrdua de dades. Admet càrregues entrelaçades (Adam7). Adequat tant per a colors plans com per a to continu (web). Desenvolupat per programadors del moviment de programari lliure.

Formats <i>bitmap</i>	Característiques
TIFF <i>tagged image file format</i> *.tiff *.tif.	Format mapa de bits. Modes RGB, escala de grisos, monotò/bitò, monocromàtic, CMYK. Admet transparència alfa (canals alfa), capes i altres recursos d'edició de la imatge en mapa de bits. Mètode de compressió LZW opcional. Adequat per a producció gràfica i impressió professional. Propietat d'Adobe Systems.
Photoshop Document *.psd. (format natiu del programa Adobe Photoshop)	Format Mapa de <i>bits</i> . Modes RGB, escala de grisos, mono/bitò i multicanal, monocromàtic, CMYK. Admet transparència alfa (canals alfa), capes i altres recursos d'edició de la imatge en mapa de bits. Adequat com a format de treball i còpia mestra en Photoshop. Desenvolupat i propietat d'Adobe Systems.
Formats vectorials	Característiques
EPS <i>encapsulated postscript</i> *.eps.	Format vectorial que admet mapes de bits amb traçats. Utilitza el llenguatge PostScript de descripció de vectors. Adequat per a producció gràfica i impressió professional. Desenvolupat per Adobe Systems. Actualment superat per altres formats com TIFF o PDF.
PDF <i>portable document format</i> *.pdf.	Format vectorial que admet mapes de bits incrustats. Conserva la informació de les tipografies. Suporta recursos d'hiperenllaços i interactivitat. Adequat per a producció gràfica i impressió professional. Adient també per a distribució i publicació digital de documents. Desenvolupat i propietat d'Adobe Systems. El 2008 va ser reconegut com a estàndard ISO.
SWF <i>shockwave Flash</i> *.swf.	Format vectorial que admet gràfics vectorials, mapa de bits incrustats, animacions i elements interactius. Requereix un connector o <i>plug-in</i> per a visualitzar-se en el web, i Flash Player per a executar-se en disc. Es visualitza integrat (<i>embedded</i>) en la pàgina HTML. Adequat per a gràfics vectorials, animacions i aplicacions interactives, tant per al web com per a reproducció directa de disc. Desenvolupat per Future Wave Software, Macromedia i Adobe. Propietat d'Adobe Systems.

Formats <i>bitmap</i>	Característiques
SVG <i>scalable vector graphics</i> *.svg.	<p>Format vectorial que admet gràfics vectorials, mapa de bits incrustats, animacions i elements interactius. Basat en XML i per tant compatible amb HTML.</p> <p>No requereix (en principi) cap tipus de connector o <i>plug-in</i> per a visualitzar-se sobre HTML.</p> <p>Adequat per a gràfics vectorials i sistemes interactius en el web.</p> <p>Representa l'estàndard obert per a gràfics vectorials a Internet. És una «recomanació» (equivalent a estàndard) del W3C des de 2001.</p> <p>Homologat i impulsat pel W3 Consortium.</p> <p>Desenvolupat per un consorci d'empreses i organitzacions.</p>

5. Programari de creació i tractament gràfic

El ventall actual de programes de creació i tractament de gràfics és ampli i divers. En les pàgines següents ens aproparem a les característiques d'alguns d'ells. En tot cas, però, allò que voldríem remarcar és la conveniència de seleccionar sempre un programa o un altre en funció del tipus de tasca que s'ha de realitzar i de les condicions de treball específiques.

Abordarem seguidament el programari gràfic segons els següents binomis comparatius:

- 1) programari de mapa de bits (*bitmap*) i vectorial;
- 2) programari orientat a impressió i orientat al web;
- 3) programari propietari i *open-source*.

Cal aclarir que cada programa es pot adscriure simultàniament a diverses categories.

5.1. Programari *bitmap* i vectorial

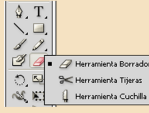

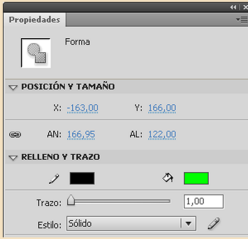
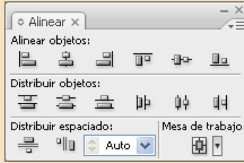

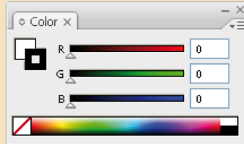
Anteriorment hem analitzat la diferenciació fonamental entre l'estructuració de gràfics en mapa de bits i vectorial. Aquesta distinció no es limita als formats, sinó que també es trasllada al programari relacionat. Hi ha, per tant, programes de gràfics vectorials que treballen amb objectes Bézier i programes de gràfics de mapa de bits que treballen amb imatges compostes per píxels.

La diferenciació entre programes *bitmap* i vectorials esdevé tan determinant que es reflecteix en el propi entorn de treball de cada aplicació. Els entorns de treball vectorial van ser concebuts, en els seus inicis, orientats al dibuix geomètric, mentre que els entorns de mapa de bits presenten un plantejament més orientat al dibuix «gestual». De fet, durant els anys vuitanta, els primers programes de gràfics divergien entre programes de dibuix (vectorials) i programes de pintura (mapa de bits). Tot i així convé no oblidar que una referència ineludible per als programes de mapa de bits ha estat l'àmbit de la fotografia.

Tot i que els programes actuals han tendit cap a una relativa hibridació enfront de la clàssica dicotomia entre treball *bitmap* i vectorial, l'imaginari de referència que va marcar els primers desenvolupaments dels entorns de treball ens ha deixat un llegat de metàfores que encara persisteix.

5.1.1. Entorns de treball vectorial

A la taula següent mostrem els elements principals de treball en un programa de gràfics vectorials.




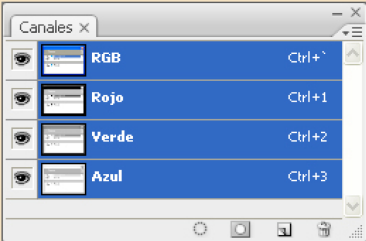
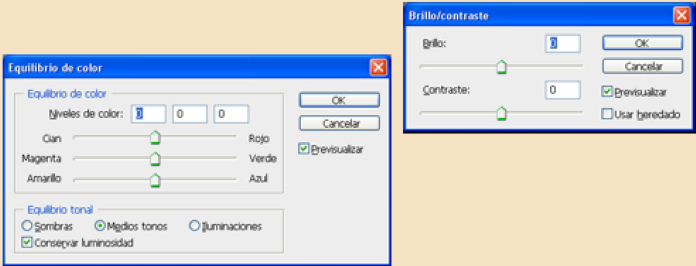
Principals elements de treball en un programa de gràfics vectorials		
Eines per a objectes Bèzier	Per a crear objectes Bèzier disposem d'una eina que crea els nodes (ploma), eines de selecció (fletxes) per a moure'ls o extraure i controlar els agafadors i d'una fulla per a tallar les línies en objectes independents. També disposem d'una eina per a crear línies rectes i d'una altra per a crear directament figures geomètriques, com rectangles, el·lipsis i polígons.	
Eines de transformació	Els objectes creats es poden transformar geomètricament. Disposem d'eines i/o procediments per a girar, desplaçar, escalar, esbiaixar o reflectir els objectes . També disposem d'opcions de menú per a combinar entre ells els objectes. La manera de fer-ho difereix d'un programa a un altre.	
Control de propietats	Els objectes són independents entre ells i tenen una sèrie de propietats que són parametrizables. D'una manera o una altra els programes ens ofereixen una forma de canviar aquestes propietats seleccionant l'objecte en qüestió i accedint-hi per mitjà d'una finestra. Les propietats es refereixen a l'amplada i al tipus de línia, al farciment de l'objecte, a la grandària i posició, etc.	
Alineació	L'alineació dels objectes respecte als altres també es pot controlar. Disposem d'un sistema de control pel qual seleccionant diversos objectes ens podem alinejar centrats, a l'esquerra, a la dreta o distribuir-los de manera que l'espai entre ells sigui igual .	
Capas	Els programes vectorials poden treballar amb capes. En cada capa hi ha un conjunt d'objectes, les capes es poden ocultar o bloquejar per a no moure algun objecte per error.	
Color	El color en un programa vectorial és una propietat de l'objecte. Podem seleccionar un objecte i assignar-li un color . També es poden crear colors nous usant els selectors de color. Però atenció: a cada color li correspon una codificació (CMYK, RGB, color de biblioteca Pantone, etc.). Cada objecte podria tenir una codificació diferent i això seria problemàtic, per exemple, si s'ha d'imprimir la imatge. És important tenir clar amb quina codificació de color es treballarà.	

Amb algunes variacions, aquests són els recursos habituals i compartits per l'entorn de treball de programes vectorials com Illustrator, Inkscape, Corel Draw o Xara Xtreme.

5.1.2. Entorns de treball *bitmap*

A la taula següent mostrem els elements principals de treball en un programa de gràfics de mapa de bits.

Principals elements de treball en un programa de gràfics de mapa de bits

<p>Eines per a dibuix i pintura</p>	<p>Les principals eines per a dibuixar píxels són el llapis, la brotxa, la goma d'esborrar, l'aerògraf i el pot de pintura que farceix àrees. A més a més, disposem d'un control de paràmetres per al llapis i la brotxa (forma, duresa, etc.). L'eina de comptagotes permet escollir un color de la imatge.</p> 
<p>Eines de selecció</p>	<p>Per a seleccionar àrees de píxels hi ha les eines de selecció, com la selecció regular (rectangle o el·lipse), el llaç per a seleccions irregulars i la vareta màgica per a seleccionar píxels similars contigus o de tota la imatge. Les eines de selecció es complementen amb una sèrie d'opcions de menú relacionades amb la selecció.</p> 
<p>Selector de color</p>	<p>El selector de color serveix per a escollir el color amb què pintar. S'ofereixen diversos sistemes per a definir el color (HSV, RGB, CIE L*a*b, etc.).</p> 
<p>Canals</p>	<p>Els programes de mapa de bits representen el color a través de canals. En cada canal hi ha un component de color. Per exemple, en el mode RGB tenim tres canals: vermell, verd i blau. És possible visualitzar solament un canal i treballar solament amb aquest component en escala de grisos.</p> 
<p>Ajustar</p>	<p>Els programes de mapa de bits tenen un referent en el retoc fotogràfic de laboratori. Per això ens ofereixen eines precises per modificar i ajustar la imatge.</p> 

Filtres
algorítmics

La disposició dels píxels en la retícula es pot modificar aplicant-los un algoritme, ja que al cap i a la fi es tracta de dades numèriques. Hi ha una gran diversitat de filtres algorítmics que creen efectes determinats en la imatge intervenint en el color dels píxels. Els programes de gràfics incorporen la possibilitat d'aplicar aquests filtres que moltes vegades són petits programes externs que s'utilitzen des del programa.

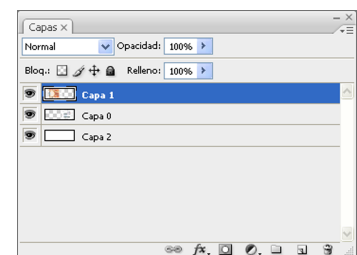
5.1.3. Confluències

Com hem vist anteriorment, el tipus de codificació de la imatge (vectorial o mapa de bits) sol condicionar el tipus d'interfície de l'entorn de treball i, per tant, la manera de treballar.

Ara bé, els programes per il·lustració vectorial han anat progressivament integrant en les interfícies elements propis dels programes de mapa de bits i a la inversa. Així doncs, tal com ja hem comentat, assistim progressivament a una major convergència de procediments de treball entre els dos tipus de programes.

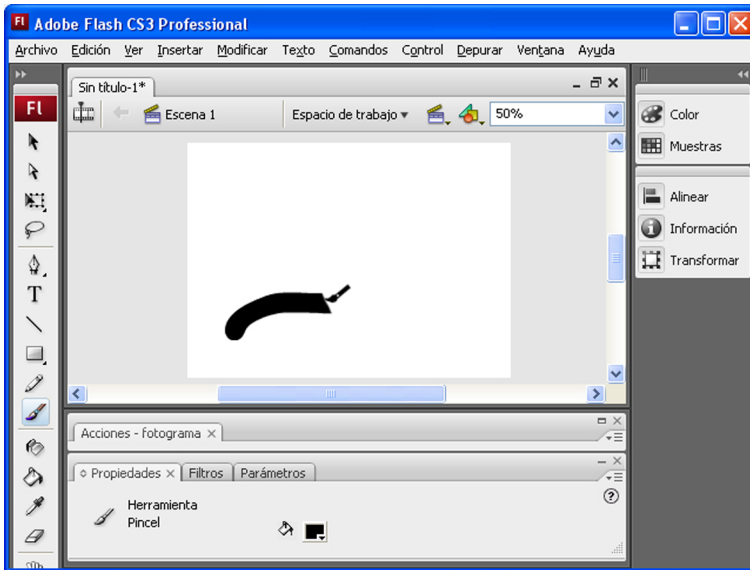
Un dels exemples més significatius és la incorporació del recurs de capes als programes de mapa de bits. Inicialment no en disposaven, però aviat van adoptar aquest recurs per influència de l'experiència de treball amb programes vectorials. I és que un sistema de capes permet preservar de manera independent qualsevol part de la imatge (com si fos un objecte vectorial). Així, podem aplicar ajustaments o filtres de manera autònoma i puntual.

En sentit contrari destaquem la incorporació als programes vectorials d'eines com el pinzell i la goma. Amb aquestes eines l'usuari pot dibuixar d'una manera més intuïtiva objectes Bézier. I és que les «línies» del pinzell no són tals, ja que en realitat l'usuari crea àrees vectorials, és a dir, objectes tancats.



Finestra de capes a Photoshop. Diferents parts de la imatge es troben en capes diferenciades. Aquesta articulació permet un treball diferenciat per elements.

Il·lustració realitzada amb el pinzell de Flash



Un cas paradigmàtic d'aquesta confluència fou (i encara és) el programa Adobe Fireworks¹⁵. En principi es tracta d'un programa d'il·lustració vectorial que, com la majoria, admet la importació de gràfics de mapa de bits. Però Fireworks va més enllà i permet generar gràfics de mapa de bits i modificar-los en el mateix programa. De fet, Fireworks integra totes les característiques pròpies tant d'un programa de mapa de bits com d'un programa d'il·lustració vectorial. La prova més representativa és la diferenciació en el seu panell d'eines d'un grup de recursos per a l'edició en mapa de bits i d'un altre grup per a l'edició vectorial. En l'entorn de treball conviuen elements vectorials amb elements de mapa de bits i la finestra de propietats és contextual i canvia en funció de l'element escollit.

⁽¹⁵⁾El maig de 2013, Adobe presentava la darrera versió, fins a la data, del seu programari anomenat Creative Cloud. Anunciava noves versions dels programes Photoshop, Dreamweaver, Flash i d'altres. Tanmateix, informava que no havia procedit a actualitzar la darrera versió de Fireworks CS6. La companyia informava d'un canvi d'estratègia que suposava la no actualització d'aquest programa que considerava que solapava amb les seves funcions els recursos d'altres programes propis com Photoshop, Illustrator o Edge Reflow.

La codificació definitiva de la imatge només serà possible en finalitzar la tasca gràfica, ja que caldrà exportar i seleccionar el format definitiu de sortida del gràfic. Mentrestant, Fireworks treballa en una versió nativa de PNG. Una finestra d'optimització permet configurar les característiques per tal d'exportar l'arxiu en format de mapa de bits.

5.2. Programari orientat a impressió i orientat al web

La finalitat dels gràfics condiciona les característiques del programa a seleccionar per a la seva edició.

El procés de producció pot ser significativament diferent si la destinació final del producte gràfic és la impressió o, altrament, un producte a distribuir en entorns digitals de visualització per pantalla. La sortida determinarà, doncs, les eines de treball del programa d'edició corresponent.

Els editors de gràfics actuals, tanmateix, ofereixen prou versatilitat per satisfer les dues necessitats alhora. Gran part de la producció gràfica del nostre temps s'orienta tant per impressió com per pantalla. Tot i així, qualsevol disseny seguirà presentant necessàriament divergències gràfiques, d'usabilitat i també tècniques pel que fa a la producció i a la posterior reproducció. Per tant, malgrat la versatilitat dels programes actuals i la transversalitat actual de l'accés al producte gràfic (multimèdia, publicació electrònica...), diferenciem seguidament algunes característiques pròpies dels programes, en funció de si el disseny s'orienta cap a la impressió o cap al web.

5.2.1. Programari orientat a impressió

Tant els programes orientats a impressió com els orientats a disseny web centren l'entorn de treball segons el format definitiu del producte final. Delimiten l'espai compositiu i ens mostren els grafismes i contragrafismes a configurar en l'interior.

Els programes orientats a impressió, tanmateix, permeten mostrar l'espai fora de marc, ja sigui per utilitzar un possible sagnat o per mostrar el conjunt d'elements de l'arxiu (pàgines, taules de treball...).

El editors de maquetació per a impressió treballen sobre la pàgina (doble pàgina en el cas de programes de maquetació editorial) o format individual (anomenades «taules de treball» a Illustrator) imprès. Aquests programes faciliten, a més, i de manera molt característica, una àrea de seguretat addicional (si així es configura), coneguda com a sagnat, contigua als «límits» de la pàgina (format refilat). Fins a aquesta zona (generalment de 3 mil·límetres) s'estendran aquells grafismes que superin el límit de tall de la pàgina. L'objectiu és evitar l'aparició d'un filet blanc no desitjat en cas d'una eventual desviació de la guillotina en el tall sobre el producte ja imprès. Igualment aquests programes coincideixen a l'hora d'oferir un panell de pàgines o d'àrees de treball per tal de visualitzar i desplaçar-se entre el conjunt elements del producte total.



Panell d'eines de Fireworks

Els programes d'autoedició per impressió treballen amb mesures físiques (centímetres, polzades, etc.) i faciliten recursos com ara guies, quadrícules i regles per tal de diagramar (i per tant estructurar) la maquetació. Igualment aquests programes, sobretot els de compaginació, permeten configurar i utilitzar el versàtil recurs de les anomenades pàgines mestres. Aquestes pàgines funcionen com plantilles que s'aplicaran a les pàgines del document que seleccionem, implementant els elements i característiques gràfiques «mestres» configurades.

D'altra banda, es disposa d'un detallat control tipogràfic que permet una configuració minuciosa de la composició de caràcter i de la composició de paràgraf. Composicions que es podran ajudar dels respectius fulls d'estil diferenciats també per cada nivell.

Cal tenir en compte que aquests programes de maquetació són d'estructura vectorial, però integren les imatges en mapa de bits respectant la seva naturallesa. Aquesta integració pot realitzar-se de forma incrustada o vinculada. Per a aquesta darrera opció resulta fonamental controlar en tot moment la ruta de vinculació de les imatges *bitmap* integrades. Tant és així, que llevat que forcem la incrustació de les imatges en el mateix document (amb el conseqüent augment de memòria de l'arxiu), qualsevol desplaçament del document sense acompanyar-lo de les corresponents imatges trencarà la vinculació i impedirà una reproducció (i fins i tot una edició) amb garanties de les imatges integrades. Per controlar aquest aspecte fonamental, els programes ofereixen un panell per a gestionar la vinculació de les imatges maquetades.

Igualment, atès el caràcter orientat a impressió, aquests programes poden treballar en diversos modes de color i permeten recórrer a paletes de selecció cromàtica RGB, CMYK i de tintes planes.

Quadricromia i tintes planes

La quadricromia separa virtualment el disseny a imprimir en quatre separacions trames que després imprimirà en màquina de forma conjunta i superposada mitjançant les corresponents tintes cian, magenta, groc i negre (CMYK).

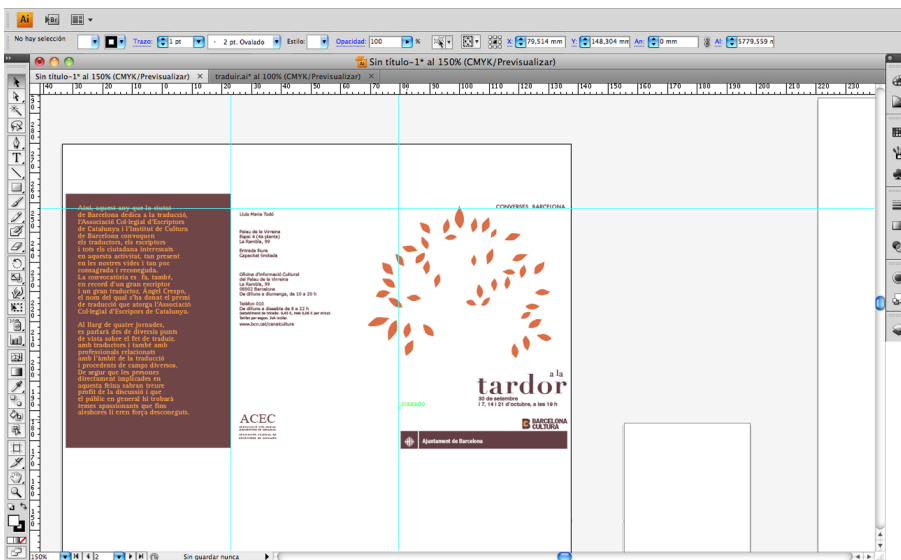
Les anomenades tintes planes o tintes directes s'utilitzen per a la reproducció directa de colors concrets (els corresponents a les tintes) en lloc o addicionalment a la quadricromia convencional (CMYK). Caldrà, però, que l'impressor corresponent disposi de les tintes directes, integrades en el disseny, per a una impressió efectiva. De fet, la biblioteca de color digital utilitzada dependrà de les tintes que realment s'utilitzin en la impressió. El més habitual en el mercat occidental per impressió òfset acostuma a ser les cartes de color de Pantone Matching System.

Atesa la previsible sortida per impressió, i la sempre crítica consistència de color que acompanya la producció gràfica impresa, aquests programes acostumen a oferir una vista de «simulació» aproximada del resultat imprès, mirant de reproduir (per avançat) els efectes de la superposició de tintes i altres atributs.

Pel que fa a la sortida, aquests programes ofereixen un control força detallat per a l'exportació del disseny final a un format com PDF. De fet, permeten diferenciar entre sortida per impressió i per pantalla. En el cas de la impressió poden configurar aspectes com l'exportació per plecs o les marques de tall i la implementació efectiva del sagnat corresponent.

Com hem apuntat, els programes orientats a impressió acostumen a ser de naturalesa vectorial, tot i que admeten gràfics de mapa de bits. Convindria, però, diferenciar els programes per a productes no necessàriament paginats (per exemple, els dissenys multiformats), com són Adobe Illustrator, Corel Draw o Xara Xtreme, de les aplicacions més orientades a la compaginació de productes editorials com Adobe InDesign, QuarkXPress o el programari obert Scribus.

Interfície de treball de l'Illustrator



D'altra banda, i pel que fa al treball orientat a la impressió, cal també esmentar els programes per al tractament d'imatges en mapa de bits com Adobe Photoshop o GIMP. Aquests programes, de fet, poden editar qualsevol imatge, ja sigui per impressió com per sortida en pantalla. I és que paràmetres com el mode de color, la resolució, el pes en memòria o el propi format de sortida podran ser detalladament configurats a través d'aquests programes.

5.2.2. Programari orientat al web

En la creació de gràfics per a sortida web o, en general, per a ser visualitzats en pantalla, les necessitats són diferents de les descrites anteriorment.

Com hem indicat, l'espai de treball d'aquests programes mostra directament el format definitiu (dins de marc). No mostren, per tant, espai de sagnat ni té sentit treballar en termes de pàgina. Per tant, tampoc utilitzen un sistema

de mesures físiques. Les unitats de mesura són, generalment, els píxels i les dimensions de l'àrea de treball es troben relacionades amb la mida final de la imatge també en píxels en el dispositiu de sortida.

Els paràmetres que cal configurar seran diferents si generem gràfics vectorials o gràfics de mapa de bits. En els gràfics *bitmap* per al web convé especialment configurar una resolució adient i un pes en memòria funcional. Tot i així, aquests programes de gràfics introdueixen controls d'optimització que permeten configurar diferents paràmetres alhora que poden reproduir (simulació), en temps real, l'efecte d'aquesta optimització en la imatge.

Pel que fa als gràfics vectorials, també per la major flexibilitat que permet no treballar amb píxels, els programes corresponents permeten introduir-hi i configurar (programar) animacions a integrar en webs o altres sortides per pantalla. Per això, programes com Flash, Silverlight o Gnash presenten un entorn de treball amb recursos com ara línia de temps, capes i fotogrames pensats per a animació i també sistemes de programació per a afegir interactivitat. Cal matissar, però, que programes com Adobe Photoshop també incorporen funcions com la línia de temps que permeten generar animacions.

Atesa l'orientació per pantalla, tots aquests programes treballen en mode RGB (i també indexat) i permeten seleccionar el color mitjançant el codi hexadecimal que utilitza HTML.

Anteriorment hem indicat alguns programes especialitzats en la generació de gràfics vectorials (amb la possibilitat d'integrar animacions, programació o interactivitat) per al web o pantalla. Cal també afegir-hi programes que permeten dissenyar per impressió, però que són perfectament útils per configurar gràfics vectorials web com Adobe Illustrator o el programari de codi obert Inkscape.

5.3. Programari propietari i *open-source*

5.3.1. El programari

Els sistemes digitals treballen sobre la base del codi binari, és a dir, instruccions codificades en zeros i uns. Però els programadors no escriuen directament en aquest llenguatge de màquina. Utilitzen els anomenats llenguatges d'alt nivell (com C, Pascal, Java, etc.). Són llenguatges lògics, amb regles sintàctiques rígides i semblances amb el llenguatge humà. La programació en aquests llenguatges d'alt nivell es tradueix posteriorment a llenguatge binari de baix nivell (llenguatge màquina) per tal que l'ordinador executi els processos corresponents. Allò que el programador escriu s'anomena codi font; allò que l'ordinador processa, codi binari.

El programari constitueix, en definitiva, un conjunt d'instruccions programades que permeten realitzar operatives de treball. El programari es pot distribuir únicament amb el codi binari per a ser utilitzat o, també, amb el codi font inclòs. I aquesta darrera possibilitat permet analitzar com ha estat programat i modificar-lo o desenvolupar-lo. La seva inclusió, doncs, marcarà una clara diferència entre programes, en relació amb l'obertura per desenvolupar el programari.

5.3.2. L'aparició del programari de propietat

En les dècades inicials de la informàtica l'incipient programari es desenvolupava a les universitats i als centres de recerca. El codi font es trobava disponible per a qualsevol desenvolupador. A començaments, però, dels anys vuitanta, amb la introducció i generalització de l'ordinador personal, els programes es comencen a comercialitzar com un producte independent de l'ordinador. Es distribueixen sense el codi font i des d'aquell moment una de les preocupacions de la creixent indústria de programari serà protegir els seus productes de la competència, mitjançant patents i la legislació de *copyright*.

En l'àmbit gràfic, a partir del llançament el 1984 de l'ordinador Macintosh, companyies de programari com Aldus amb Pagemaker, Quark Inc. amb QuarX-Press, Macromedia amb Freehand i Adobe amb Photoshop s'introdueixen en els estudis de disseny per consolidar-se com a imprescindibles. Aquest programari «revolucionarà» el mètode de treball de dissenyadors, però també de tota la indústria gràfica en general. L'anomenada autoedició, però, no serà més que el primer pas d'una trajectòria que ben aviat amb Internet i el desenvolupament multimèdia situarà els programes i les eines relacionades en el centre d'interès de la indústria, la investigació i la comunitat d'usuaris i creadors.

El 1994 Adobe absorbirà la competidora Aldus (desenvolupadora de la pionera aplicació de maquetació PageMaker), i el 2005 farà el mateix amb Macromedia, la propietària de programes com Freehand, Dreamweaver, Flash o Fireworks. Adobe Systems constitueix, doncs, en el mercat actual un desenvolupador fonamental, propietari d'un ampli ventall d'aplicacions que són referents en el disseny, l'autoedició i el multimèdia.

La comercialització que faran totes aquestes companyies del seu programari serà, per tant, sense acompanyar-lo del codi font. Qualsevol modificació o desenvolupament del programa per actors aliens a l'empresa propietària quedarà potencialment limitat.

5.3.3. El projecte GNU

A mitjan anys vuitanta un grup de programadors inicien un projecte per tal d'«alliberar», segons les seves paraules, el programari. Richard M. Stallman, que havia treballat en el MIT, en el desenvolupament del sistema operatiu UNIX i eines relacionades, inicia el projecte GNU (acrònim de *GNU's not UNIX* i traduïble com a *GNU no és UNIX*). L'objectiu serà crear un nou sistema operatiu equivalent a l'UNIX però d'ús i desenvolupament lliure. La tasca és complexa perquè les eines disponibles que necessitaven (programes de text, compiladors, etc.) també eren propietàries. Així que serà precisament des d'aquest punt per on començaran. Simultàniament crearan la Free Software Foundation (FSF), organització a la qual conferiran l'encàrrec de vetllar pel procés.

5.3.4. La llicència GPL

Un dels resultats del projecte GNU, en relació amb la legislació del *copyright*, és la creació de la General Public Licence (GPL). Aquesta llicència parteix dels drets d'autor i de reproducció per a definir els termes sota els quals es distribuirà un programa. Un programari amb llicència GPL s'ha de distribuir conjuntament amb el seu codi font. Igualment la llicència permet la lliure distribució i modificació del codi, sempre que tota modificació del programari resti igualment i de forma obligatòria sota la mateixa llicència GPL. Qualsevol desenvolupament, doncs, basat en programari sota llicència GPL continuarà essent lliure per no infringir la llicència que permet la seva distribució.

5.3.5. El moviment del programari lliure

Actualment el moviment que desenvolupa programari lliure de codi obert (*open-source*) disposa al seu abast d'un sistema operatiu (Linux) que funciona en la majoria de servidors d'Internet i compta amb múltiples eines de programari.

Companyies importants del sector (com IBM, Sun o Novell) contribueixen a aquest procés, així com un important nombre de petites empreses, programadors independents i equips informals. Empreses relacionades amb Internet com Google també participen en el finançament de projectes (amb el *Google Summer of Code*, des de 2005).

A nivell de programari cal remarcar la tasca de la companyia Xara, desenvolupadora del programari vectorial Xara Xtreme, la Blender Foundation, que coordina el desenvolupament del programari d'animació i gràfics 3D Blender, o dels diferents equips de desenvolupament de programes com ara GIMP, Inkscape o Scribus.

5.3.6. Programari lliure per a gràfics

Actualment podem accedir a programes de codi obert tant per al tractament de gràfics en mapa de bits com per a l'edició vectorial. Aquests programes es troben disponibles per a plataformes Linux, Windows i Macintosh.

GIMP (*GNU Image Manipulation Program*) és un aplicatiu per a imatges *bitmap* amb funcionalitats bàsiques equivalents a les facilitades per Adobe Photoshop. Pel que fa a l'edició vectorial, podem trobar programes com ara Inkscape i Xara Xtreme per a l'edició pròpiament de gràfics vectorials, o Scribus per al desenvolupament de maquetacions.

Els defensors del programari propietari argumenten que un desenvolupament d'eines com el que s'ha produït en el món gràfic en les darreres dècades solament és possible amb una indústria forta de programari. I aquesta es finança amb la comercialització de llicències de programes. Per la seva banda, des dels sectors afins al programari lliure s'argumenta que moltes vegades aquesta indústria inverteix més esforços a oferir noves funcionalitats (no sempre prou necessàries) als usuaris, a través de noves versions (comercialitzables), que no pas a corregir els errors de les versions vigents dels programes.

Amb l'adquisició de Macromedia per part d'Adobe el 2005, la companyia nord-americana es va convertir en l'actor principal en el desenvolupament de programari propietari en l'àmbit gràfic. Programes com Photoshop, Flash, Illustrator, InDesign o Dreamweaver, entre d'altres, són propietat d'Adobe. La companyia comercialitza els seus programes alhora que, especialment a partir de 2008 amb l'anunci del seu *Open Screen Project*, inicia una política d'obertura de les especificacions dels seus formats SWF, FLV o PDF.

Per altra banda, companyies nascudes amb Internet com Google segueixen introduint-se en el desenvolupament d'aplicatius gràfics, començant per eines orientades a l'usuari.

El desenllaç i evolució, en tot cas, d'aquesta situació dependrà de les actuacions dels diversos agents implicats i, en bona mesura, de l'actitud dels usuaris.

Bibliografia

Bibliografia

Blasco, L. (2011). *Sobreimpresión de la pantalla al papel y viceversa*. Barcelona: Index Books.

Gatter, M. (2011). *Manual de impresión para diseñadores gráficos*. Barcelona: Parramón Ediciones.

Gordon, B.; Gordon, M. (2007). *Manual de diseño gráfico digital*. Barcelona: Gustavo Gili.

Johansson, K.; Lundberg, P.; Ryberg, R. (2007). *Manual de Producción Gráfica. Recetas (Segunda edición actualizada y ampliada)*. Barcelona: Gustavo Gili.

Nickel, K. (ed.) (2011). *Ready to Print. Handbook for media designers*. Berlín: Gestalten.

