

Laboratorio Online Permanente di Tecnologie Internet per la Scuola – #loptis

Un cMOOL: connectivist Massive Open Online Laboratory

FEBRUARY 23, 2014

Elaborazione di immagini – tre fatti che fanno la differenza – #loptis Reply

#loptis • Tags: compressione conservativa, compressione non conservativa, free software, GIF, Gimp, immagini bitmap, immagini raster, immagini vettoriali, Inkscape, JPEG, pdf, PNG, RAW, software libero, SVG, XCF, ZIP

- *Sicuramente un post per gli studenti di “Editing multimediale” della IUL ma anche per tutti quelli a cui capita di lavorare con le immagini*
- *Grafica bitmap e vettoriale*
- *Due o tre cose sui formati più noti, GIF, PNG e JPEG, sulla compressione delle informazioni, conservativa e non, e sui formati “dedicati” XCF e SVG*
- *Gimp e Inkscape, due bellissimi software liberi per l’elaborazione delle immagini che girano su Linux, Mac e Windows*
- *Elaborazione delle immagini in livelli*

Primo fatto: software libero anziché proprietario

Il fatto che il software sia libero o proprietario è fondamentale, per motivi etici prima ancora che tecnici. L’espressione sintetica di Marina (<http://iamarf.org/2013/09/22/iniziando-ad-accordare-i-suoni-loptis/#comment-23528>) è perfetta:

software libero vuol dire che rende le persone più libere

Aggiungo: la scuola e l’università hanno il dovere morale di dimostrare, diffondere e promuovere il software libero. Il resto l’abbiamo detto in un post precedente ([il-software-libero-ti-libera-loptis](#)).

Secondo fatto: distinguere fra grafica bitmap e vettoriale

Immagini digitali

Le immagini digitali sono sempre formate da una sorta di scacchiera di piccoli quadratini denominati pixel, il cui interno è rappresentato con un colore uniforme al quale nel computer corrisponde un certo valore numerico. Anche se le immagini sono monocromatiche vale lo stesso concetto. Per fare un esempio noto a tutti, le immagini TAC (Tomografia Assiale Computerizzata) sono rappresentate in scale di grigi. Ebbene, anche lì in ogni pixel, l'intensità di grigio è in relazione con un numero che rappresenta (con una certa approssimazione) la densità media dei tessuti del corpo nella spazio rappresentato da quel pixel.

dpi e risoluzione

In qualsiasi immagine digitale potete scorgere i pixel, se vi avvicinate o la ingrandite a sufficienza. Naturalmente è proprio quello che non si vuole vedere, di solito. In questo articolo useremo il termine *risoluzione* per il numero di pixel dell'immagine – maggiore risoluzione significa maggior numero di pixel. In realtà, il concetto di risoluzione spaziale di un'immagine è differente e molto più complesso ma qui ci atterremo alla prassi corrente per rendere il discorso più agile e compatto. Invece con *dimensione* intenderemo la grandezza del file in cui l'immagine viene memorizzata, e potrà essere espressa in Mega Byte (MB) o in Kilo Byte (KB) [1]. Per esempio se fate una scansione di un documento o un'immagine dovete fare attenzione al parametro dpi (https://it.wikipedia.org/wiki/Punti_per_pollice) (dots per inch, punti per pollice). Un pollice sono 2.54 cm e valori accettabili stanno usualmente fra 100 e 200 dpi, ma dipende da cosa dovete scannerizzare. Occorre quindi fare delle prove scegliendo il valore più basso che dà la qualità adeguata, nel contesto che vi interessa. Non vale esagerare, perché la dimensione delle immagini cresce grandemente in proporzione inversa alle dimensioni dei pixel: se i pixel di un'immagine sono più piccoli vuol dire che ve ne saranno di più – se scelgo 200 dpi anziché 100, vuol dire che su ogni lato dell'immagine ho il doppio dei pixel, quindi il numero totale dei pixel dell'immagine sarà $2 \times 2 = 4$ volte più grande, ergo raddoppiare la dimensione lineare vuol dire quadruplicare la dimensione dell'immagine, e quindi anche la dimensione del file dove questa viene memorizzata. Infatti, dal punto di vista del computer, un'immagine digitale è semplicemente una fila di numeri, tanti quanti sono i pixel in cui è suddivisa. Quando il computer la deve rappresentare su un qualche supporto – monitor, stampa... – allora legge i numeri a partire dal primo e decodifica ciascuno di essi in maniera da produrre l'effetto dovuto, in termini di colore e intensità, poi li "scrive" nello spazio dell'immagine come si fa nelle lingue occidentali: da sinistra a destra, dall'alto al basso.

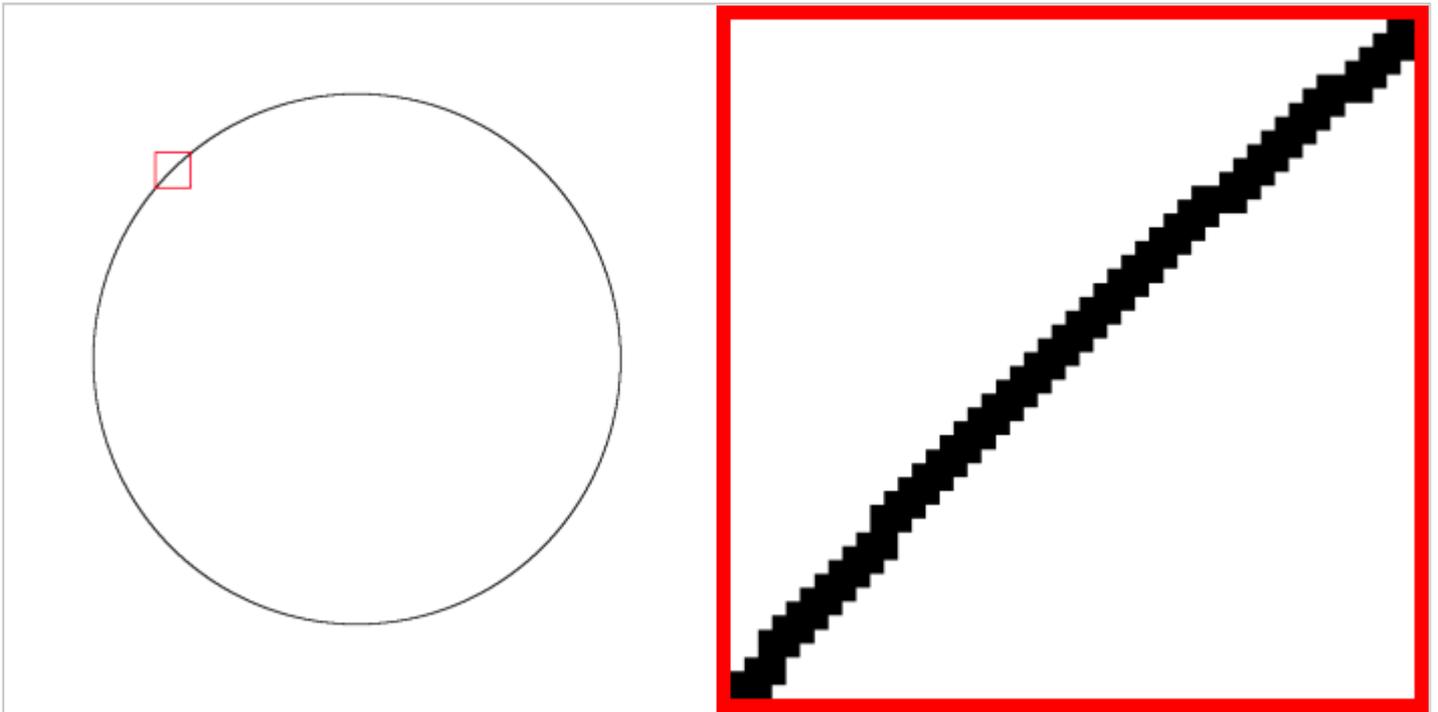
Tipi di grafica

Ma non sempre le immagini vengono memorizzate secondo la logica di un "elenco dei contenuti dei pixel". Si danno infatti due possibilità, una che prende il nome di *grafica vettoriale* (raster: griglia) e l'altra di *grafica bitmap*, o *grafica raster* (da qui in poi utilizzeremo il termine bitmap). Le immagini bitmap sono memorizzate così come abbiamo già detto. Ad esempio le fotografie e le immagini mediche sono di questo tipo. Le immagini vettoriali sono invece memorizzate attraverso una

rappresentazione ad oggetti. I disegni CAD (Computer Assisted Design) fatti da un progettista e i documenti PDF sono esempi di immagini vettoriali. Facciamo un esempio affinché la differenza sia chiara.

Grafica bitmap

Supponiamo prima di disegnare una circonferenza con un software di tipo bitmap, poi immaginiamo di ingrandire molto l'immagine: prima o poi la circonferenza rivelerà la struttura in pixel, che le darà un aspetto frastagliato.

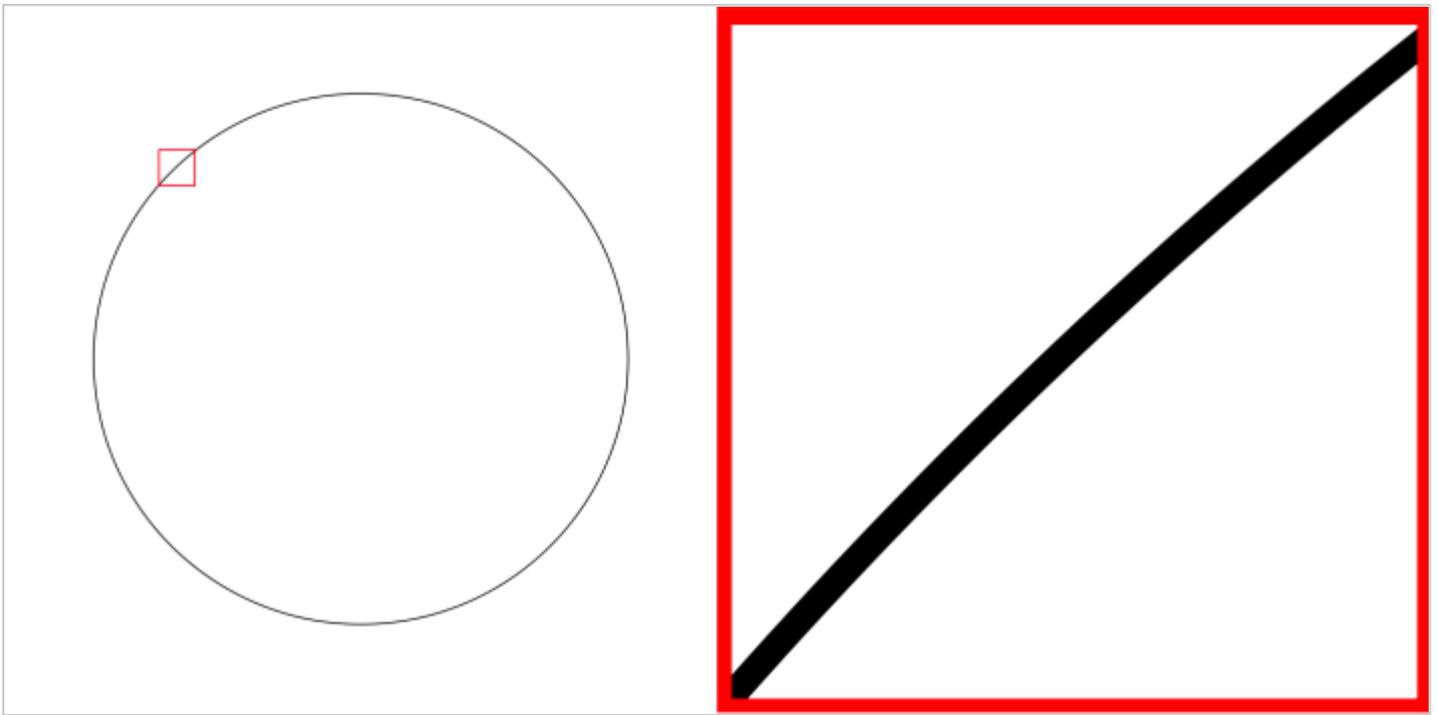


(<http://iamarf.files.wordpress.com/2014/02/circ-bitmap.png>)

Immagine di una circonferenza generata in grafica bitmap. L'immagine di destra rappresenta l'ingrandimento del quadrato rosso raffigurato nell'immagine di sinistra.

Grafica vettoriale

Disegnate ora la stessa circonferenza con un software di tipo vettoriale e provate a ingrandire l'immagine a piacimento: la circonferenza sarà sempre lì, con gli stessi identici attributi che le avrete assegnato all'atto della sua creazione. Magari ne vedete solo un piccolo arco perché avete ingrandito l'immagine veramente tanto, ma non vi sono artefatti.



(<http://iamarf.files.wordpress.com/2014/02/circ-vect.png>)

Immagine di una circonferenza generata in grafica vettoriale. L'immagine di destra rappresenta l'ingrandimento del quadrato rosso raffigurato nell'immagine di sinistra.

Perché questa differenza? Il software di tipo bitmap parte da un'immagine che è tanti pixel alta e tanti pixel larga. Dopodiché qualsiasi cosa facciate, il software si preoccupa di riempire i pixel come dite voi. Altro non fa. Se avete chiesto di disegnare una circonferenza, il sistema per utilizzare l'equazione matematica della circonferenza

$$x^2 + y^2 - 2\alpha x - 2\beta y + \alpha^2 + \beta^2 - r^2 = 0$$

dove α e β sono le coordinate del centro e r è il raggio della circonferenza, per riempire appropriatamente i pixel. Poi se ne dimentica, ovvero dimentica i suoi parametri, α , β e r [2]. Invece, il software di tipo vettoriale, utilizza l'equazione della circonferenza per disegnarla sullo schermo, con i parametri di rappresentazione correnti, ovvero riempie i pixel che voi vedete sullo schermo, e invece di memorizzare i contenuti memorizza l'equazione della circonferenza, sotto forma dei suoi parametri α , β e r , e di altri eventuali attributi grafici, quali spessore e colore della linea. Poi, ogni volta che voi rinfrescate l'immagine sullo schermo, ricaricandola o variandone le dimensioni, ricalcola i contenuti dei pixel a partire dall'equazione della circonferenza. E così per tutti gli altri oggetti.

Riassumendo, con la grafica bitmap il sistema ricorda i contenuti di tutti i pixel dell'immagine – quando la deve rappresentare sul schermo costruisce l'immagine semplicemente rappresentando adeguatamente i singoli pixel. Con la grafica vettoriale il sistema ricorda l'idea di circonferenza, ovvero di quella precisa circonferenza con il centro posizionato in quella certa posizione, con quel certo raggio, con quel certo spessore e quel certo colore con cui deve essere tracciata – quando deve rappresentare l'immagine, prima calcola i contenuti dei pixel per rappresentare quella circonferenza, poi rappresenta i pixel così calcolati sullo schermo.

Formati – compressione

Non vogliamo perderci qui nella foresta dei formati. Se necessario approfondiremo dove opportuno.

Solo gli esempi comuni. Tutte le immagini di origine fotografica sono gestite e memorizzate con la logica bitmap. È naturale: derivano tutte da una qualche matrice di rivelatori, intendendo per matrice una qualche precisa disposizione di rivelatori di radiazione elettromagnetica: luminosa, infrarossa, X o altro. I formati più comuni per la distribuzione di immagini bitmap nel web sono **GIF**, **PNG** e **JPEG** – detto anche **JPG**, è lo stesso.

Qui occorre un inciso sulla compressione. Le immagini – anche l'audio e a maggior ragione il video – contengono molta informazione, quindi i file tendono ad essere grandi – oggi si ama dire che sono “pesi”. Persino la fotocamera di un telefono produce immagini di vari milioni di pixel. Far viaggiare questa roba in internet può essere un problema, a maggior ragione per chi vive in paesi sottosviluppati – o “diversamente sviluppati”, come il nostro, duole dirlo ma all'atto pratico, fra la velocità di trasferimento subalpina e quella sovralpina può corre tranquillamente un fattore 10. Le tecniche di compressione dell'informazione sono quindi fondamentali, è grazie a queste che la distribuzione dell'informazione multimediale è esplosa, ad esempio nei formati mp3 (audio), jpeg (immagini) e mp4 (video). Tutti questi esempi – sono i più comuni ma ve ne sono tanti altri – sono caratterizzati dal fatto di impiegare un qualche sistema di compressione dell'informazione. E tutti e tre usano i sistemi di compressione più efficaci, ovvero quelli **non conservativi**, che per comprimere meglio buttano via qualcosa – qualcosa che non si sente nell'audio, che non si vede nelle immagini o nei video. Sono sistemi regolabili – se si esagera si finisce col deteriorare il messaggio.

Con la **compressione conservativa** invece non si butta via nulla, ma si sfruttano ripetizioni e regolarità, sfruttandole in modo intelligente. Ci capiamo subito con questo esempio: pensate a un'immagine fatta da un solo pixel centrale nero e tutti gli altri bianchi – non è interessante ma è utile per capire. Non è difficile immaginare un sistema abbastanza furbo da memorizzare il numero di pixel che sono uguali e il loro (unico) valore. Nel nostro esempio significa che invece di memorizzare milioni di numeri, basta memorizzare il numero dei pixel bianchi, il numero di quelli neri e i due valori corrispondenti al bianco e al nero: 4 numeri! Abbiamo semplificato molto ma il concetto c'è, ed è anche sufficiente a capire che il successo di questi metodi dipende dal tipo di immagine. Per esempio in un'immagine fotografica piena di sfumature morbide ci sarà poco da comprimere. Un esempio di compressione conservativa molto noto è quello del formato **ZIP**. Può valer la pena di fare un giochino. Prendo un brano di questo scritto, lo salvo in un file con un editore di testo – **un editore di testo, non Word!** – Mi è venuto un file di 1075 byte. Sempre con l'editore di testo scrivo “pippo pippo...” tante volte (alla Shining – non mi sta vedendo nessuno...) fino a ottenere un file egualmente lungo, ovvero di 1075 byte. Applico a tutti e due lo zip: nel primo caso viene un file di 725 byte e nel secondo di 173 byte. Potete provare anche voi per esercizio, inventandovi degli esempi simili, o scaricando il [file numero 1](http://lai.dfc.unifi.it/lab/p1) (<http://lai.dfc.unifi.it/lab/p1>) e il [file numero 2](http://lai.dfc.unifi.it/lab/p2) (<http://lai.dfc.unifi.it/lab/p2>) per zipparli da voi.

Dunque dicevamo che i formati più comuni per la distribuzione di immagini bitmap nel web sono, GIF, PNG e JPEG. Vediamoli brevemente.

Il formato GIF usa una compressione conservativa. Questo lo rende adatto a immagini che contengono testo o grafica composta da linee. Si può usare per fare immagini animate.

Anche il formato PNG comprime l'immagine in modo conservativo e quindi anche questo è adatto nei casi cui sono presenti testo o linee. È meglio rispetto al GIF perché è più recente: consente di rappresentare le immagini con maggiore qualità e le comprime meglio. Non si possono però fare immagini animate, per questo ci vuole il formato GIF. In fondo vedremo un esempio.

Il formato JPEG applica invece una compressione non conservativa. È il formato adatto alle immagini fotografiche. Se siamo noi a generare un file JPEG, allora all'atto del salvataggio si può scegliere il grado di compressione. Occorre controllare quindi: più si comprime e più l'immagine può risultare degradata, specialmente se vi sono dettagli fini e contrasti bruschi.

Per quanto riguarda la grafica vettoriale occorre certamente citare il **PDF**. È un'ottima occasione per puntualizzare che il **PDF è un formato grafico, e per di più vettoriale**, perché molti scambiano il PDF per una variante del formato DOC – quello di Word, o altri word processor. No, il PDF non è un formato di testo o testo formattato, bensì è un formato grafico: esportare un documento qualsiasi – testo, foglio di lavoro o altro – in PDF vuole dire farne una sorta di fotografia, ovvero congelarlo in un'immagine, e per di più un'immagine vettoriale, perché così risulta meno sensibile alla modalità con la quale viene rappresentato, grazie al meccanismo ad oggetti che abbiamo visto. Perché il PDF è nato proprio per diffondere documenti nella rete, in maniera che questi rimangano inalterati, indipendente dal supporto di visualizzazione o dalla modalità di stampa.

Manipolando foto

La questione del formato è particolarmente importante quando siamo noi che creiamo le immagini. Le immagini si possono presentare in una grande varietà di forme, agli estremi abbiamo le fotografie da un lato e i “disegni al tratto” dall'altro – diagrammi, grafici. Di foto ne produciamo tutti tante, anche troppe. La tendenza è scattare e riempire memorie – pensa a tutto la macchina. In realtà anche in un cellulare di fascia bassa si può intervenire su alcuni parametri, per non parlare degli smartphone. Sicuramente si possono determinare numero di pixel e qualità dell'immagine. La dimensione dei file cresce con il numero dei pixel e la qualità dell'immagine. La qualità è di fatto regolata mediante il rapporto di compressione **JPEG**. Prima di usare l'apparecchio, non sarebbe male fare delle prove, ripetendo lo stesso scatto con risoluzioni e qualità diverse, per poi scegliere i parametri che consentono di ridurre le dimensioni dei file ma senza introdurre artefatti nell'immagine.

Chi ha qualche velleità fotografica e utilizza macchine “serie” – reflex, *street photography*, grandi formati – ha certamente a disposizione l'opzione del formato cosiddetto **RAW - raw data: dati grezzi**. Questi sono i numeri letti direttamente dai sensori dell'apparecchio, sui quali si può lavorare successivamente con gli appositi software di elaborazione fotografica. È il formato prediletto da chi vuol partecipare alla costruzione dell'immagine fotografica, un po' come coloro che amavano sviluppare le proprie foto nella camera oscura. Con il formato RAW la libertà di espressione è maggiore ma anche la dimensione dei file è molto più grande dei file JPEG – meno scatti, più riflessione post processing – ex camera oscura – più qualità.

Fabbricando immagini

Diverso è il caso in cui le immagini ce le dobbiamo fabbricare “a mano”. Qui la risposta non è univoca perché dipende da quello che si deve fare. Nel caso di diagrammi o grafici, quasi certamente conviene optare per un formato di tipo vettoriale. Costruendo invece immagini composite nelle quali gli elementi fotografici sono importanti, allora probabilmente si può lavorare sia in forma vettoriale che bitmap – la scelta dipende dal tipo di fruizione dei risultati e, in ultima istanza, dalle preferenze personali.

Lavorare in forma grafica bitmap o vettoriale vuol dire usare software diversi. Qui citiamo due ottimi software liberi: Gimp (<http://www.gimp.org>) per la grafica bitmap e Inkscape (<http://www.inkscape.org>) per la grafica vettoriale. La quantità e varietà di software e di servizi web per l'elaborazione delle immagini è impressionante. Molti conosceranno Photoshop per il ritocco delle immagini fotografiche, e forse avranno sentito nominare Autocad per la progettazione, quali esempi di software per grafica bitmap da un lato e vettoriale dall'altro, anche se quest'ultimo più di nicchia e strettamente professionale. Sono programmi dal prezzo elevato, probabilmente

ingiustificato per la maggioranza delle finalità che può avere un utente generico. In realtà le alternative libere Gimp e Inkscape sono molto sofisticate e possono essere usate con profitto anche in contesti professionali. Gli esempi precedenti, dove abbiamo mostrato cosa succede ingrandendo immagini dei due tipi sono stati realizzati con questi due software, ma vediamo qualche esempio per dare un'idea delle potenzialità e delle differenze fra le due modalità.

Lavorare in grafica bitmap con Gimp



(<http://iamarf.files.wordpress.com/2014/02/dampfloko-full-row.jpg>)

Questa immagine – chi vuole sapere di che si tratta può leggere la nota [3] – è stata scattata con un apparecchio comune, di quelli automatici, nemmeno più tanto recente, e ciò nonostante già in grado di sparare immagini di 4320×3240 pixel, come questa che vedete qui sopra, espressa in tre diversi formati, da sinistra a destra: PNG – compressione conservativa, cioè nessun degrado ma il file occupa 19MB; JPEG con compressione ridotta (94% di qualità secondo Gimp [4]), e già così si ottiene un file di soli 3 MB; JPEG con compressione elevata, (10% di qualità sempre secondo Gimp), e si arriva a 213 KB: un fattore di compressione dell'1% e la qualità sembra la stessa! In effetti, con così tanti pixel si può andar giù pesanti con la compressione JPEG.

Ma la differenza c'è, dipende da come si intende usare l'immagine, in particolare da quanto si pensa di ingrandirla. Le tre immagini sopra sono scalate per farle entrare nel post, cliccando sopra appaiono le versioni originali, sulle quali potete cliccare un'altra volta per ingrandirle ulteriormente. Se con il vostro browser non funziona così, potete scaricarle sul vostro computer e poi ingrandirle con il software di visualizzazione di immagini che usate di solito. Ebbene, vedrete che, ingrandendo molto, se fra le versioni PNG (a sinistra) e JPEG 94% (quella di mezzo) non ci sono differenze apprezzabili, la versione JPEG 10% (a destra) risulta marcatamente degenerata, con una sorta di artefatto a blocchi. Ebbene quello è l'effetto di perdita di informazione, causato da una compressione JPEG troppo aggressiva. Dipende quindi da come si vogliono utilizzare le immagini.

Ma vediamo ancora più in dettaglio come stanno le cose, confrontando un particolare dell'immagine senza alterazioni, con risoluzione ridotta, con compressione JPEG e con tutti e due i trattamenti.



(<http://iamarf.files.wordpress.com/2014/02/dampfloko-small-row-part1.png>)

1) immagine originale ad alta risoluzione 4320×3240 pixel, dimensione 19 MB; 2) sempre ad alta risoluzione

*ma molto compressa con JPEG, dimensione 213 KB; 3) a risoluzione ridotta del 20%, 864*648 pixel, dimensione 211 KB; 4) sia a risoluzione ridotta che molto compressa con JPEG, dimensione 23 KB. Clicca l'immagine per vedere gli effetti, poi clicca ancora sull'immagine che viene per zoomarla e vedere bene gli effetti.*

Per vedere bene gli effetti occorre cliccare sull'immagine qui sopra, che è scalata per farla entrare nell'impaginazione, e poi cliccare nuovamente sull'immagine che appare per ingrandirla ulteriormente.

Ebbene, se la dimensione del file dell'immagine originale era di 19 MB, sia riducendo la risoluzione del 20% che applicando una compressione JPEG energica – 10% secondo Gimp – questa si riduce a circa 200 KB, ovvero dell'1%, e in ambedue i casi si osservano delle degenerazioni, che sono però di natura diversa. Applicando infine ambedue i trattamenti, si riduce sì di un ordine di grandezza la dimensione del file (23%) ma addio qualità.

Quindi, primo messaggio: per ridurre il “peso” di un'immagine si può agire sul parametro di qualità-compressione del formato JPEG ma occhio alla risoluzione: se i pixel sono pochi il rischio di rovinare l'immagine è alto, quindi bisogna essere prudenti: andare per tentativi e errori.

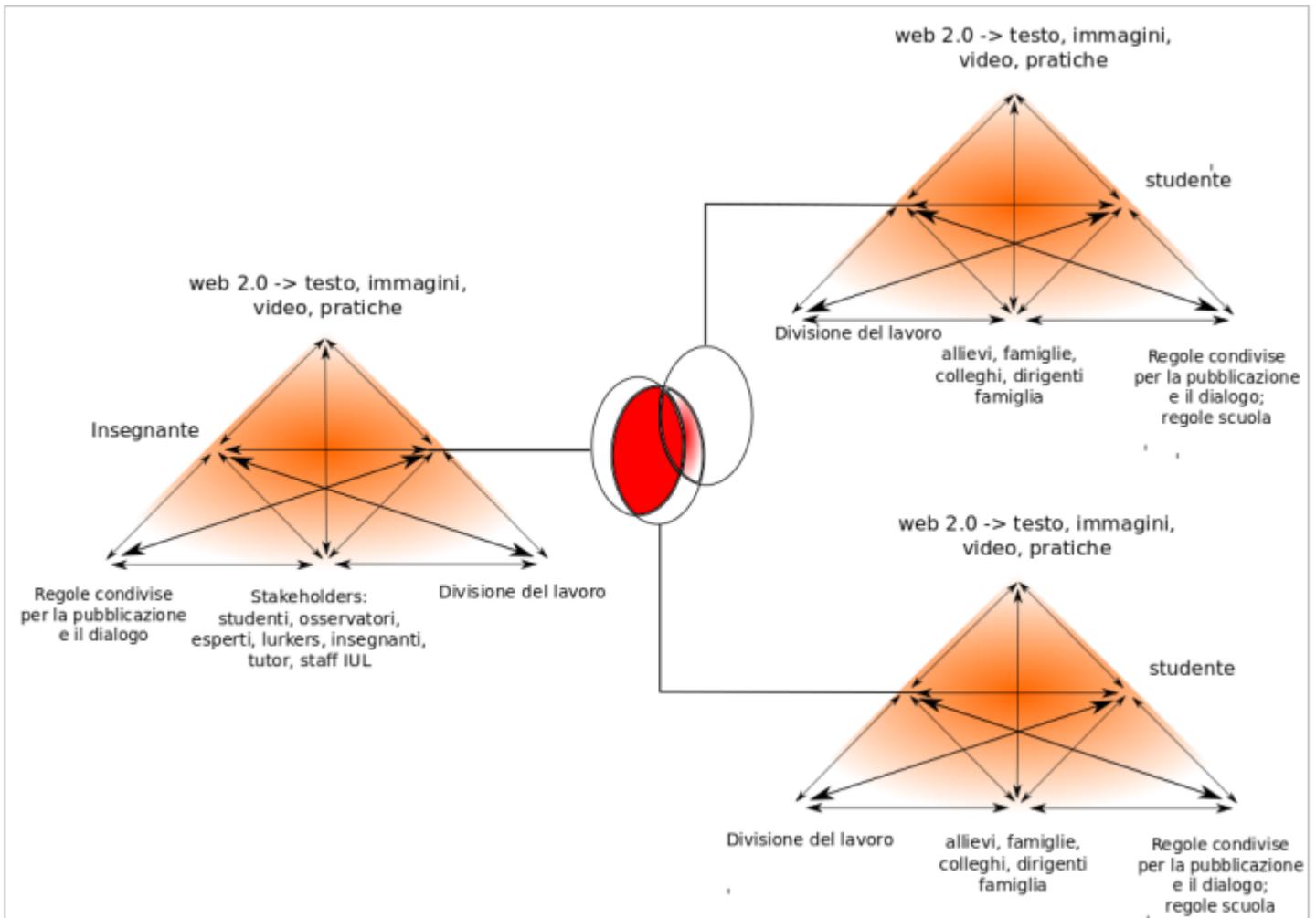
Secondo messaggio: per lavorare bene, può avere senso porsi il problema se convenga ridurre le dimensioni riducendo la risoluzione oppure comprimendo di più con il JPEG. Si confrontino le immagini 2), compressa con JPEG, e 3), alleggerita riducendo la risoluzione. La prima ha i contorni più conservati mentre si sono praticamente perse le sfumature delle superfici piane. Nella seconda le sfumature delle superfici sono ancora ben rappresentate ma i contorni sono deteriorati dalla dimensione dei pixel – meno pixel vuol dire pixel più grossi. Da tutto questo si vede come in ambedue i casi la riduzione delle dimensioni dell'immagine sia stata ottenuta al prezzo della perdita di una certa quantità di informazione, ma che tale perdita è qualitativamente diversa nei due casi. La riduzione della risoluzione tende a sciupare i contorni, la compressione JPEG tende a sparpagliare la degenerazione nell'immagine. L'indicazione pratica che se ne ricava è che nel caso di immagini morbide e con poche variazioni brusche di contrasto conviene ridurre la risoluzione, ovvero il numero di pixel; invece nel caso di immagini dove predominano linee e contorni forti, può convenire ricorrere ad una compressione JPEG più energica. In ogni caso: domandarsi come verrà usata l'immagine e cosa vogliamo comunicare e poi andare per tentativi ed errori, perché gli effetti cambiano a seconda del tipo di immagine.

Chi si cimenta nell'uso di Gimp (<http://www.gimp.org>) si accorge che fra i 40 formati nei quali può salvare un'immagine, il primo che viene proposto è **XCF**. Questo è il formato nativo di Gimp, ovvero quello che è in grado di memorizzare tutte le informazioni fra una sessione di lavoro e l'altra. È il **formato da usare** quando si lavora con un'immagine e la vogliamo mantenere per ritornarci a lavorare in futuro – se la memorizziamo in questo formato siamo sicuri di ritrovare tutta quello che abbiamo fatto in precedenza. **Non è il formato da usare** per inviare un'immagine in rete, a meno che uno non voglia condividerla con qualcun altro che debba continuare a lavorarci con Gimp. Il processo corretto quindi è quello di memorizzare tutte le varie versioni dello sviluppo in XCF e poi esportare il risultato in un formato idoneo a essere trasmesso, per esempio GIF, PNG, JPEG o altro.

Resterebbe da dire qualcosa sulle elaborazioni in grafica bitmap, ma vedremo meglio dopo, ragionando del terzo fatto che fa la differenza, in materia di manipolazione di immagini.

Lavorare in grafica vettoriale con Inkscape

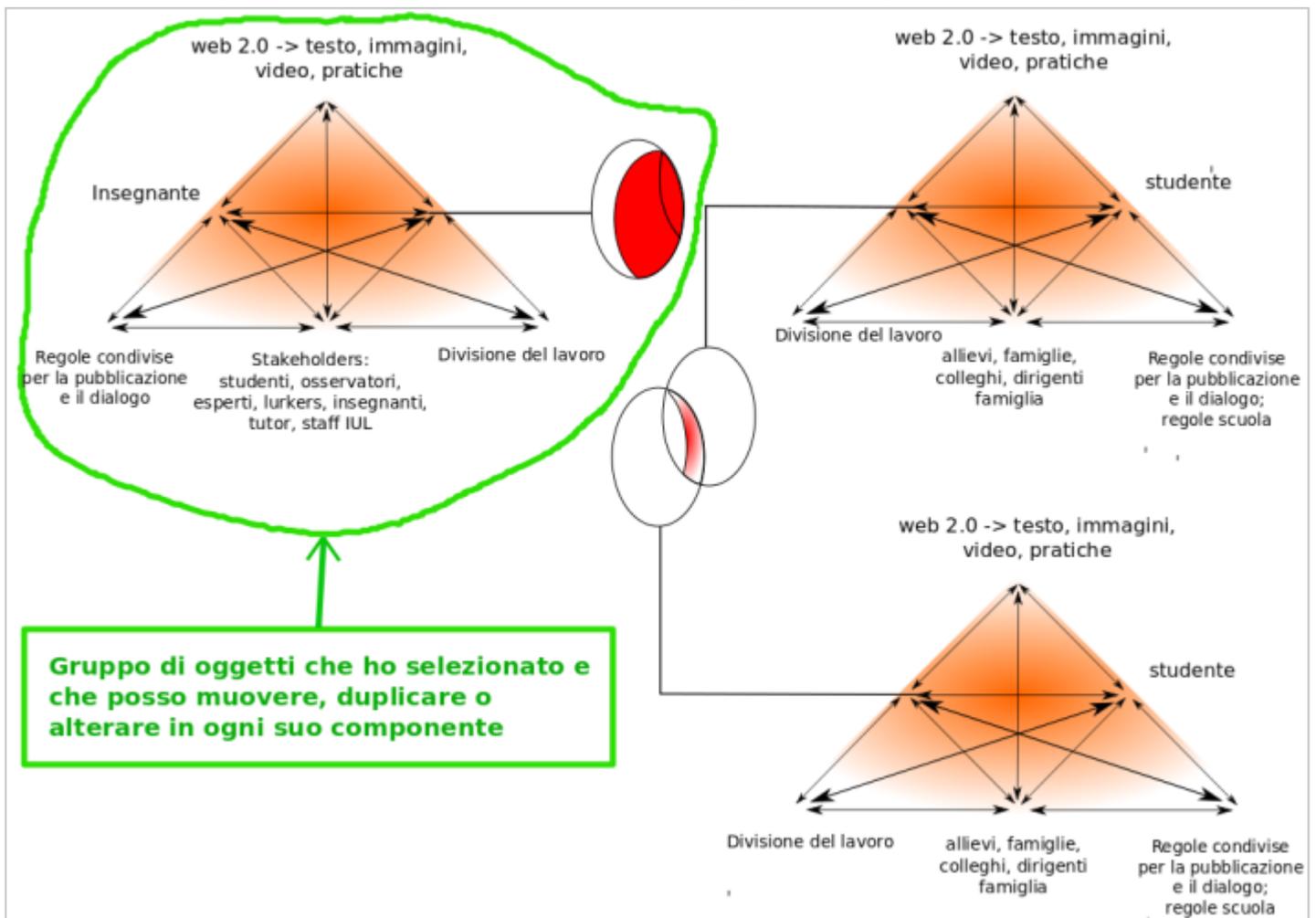
Le immagini si possono creare anche in grafica vettoriale. Anche per questo tipo di elaborazione esiste uno splendido software libero, che è Inkscape (<http://www.inkscape.org>). Vediamo un



<http://iamarf.files.wordpress.com/2014/02/activity-system-5.png>

Esempio di tipica immagine che conviene produrre in grafica vettoriale. Questa è stata prodotta con il software libero Inkscape. Clicca l'immagine per vederla meglio.

Il soggetto dell'immagine non ci interessa qui, se non per come è stato costruito graficamente. Chi fosse incuriosito dal senso del diagramma può leggere una breve nota [5]. Come abbiamo già visto, nella modalità vettoriale, ad essere memorizzati non sono i pixel dell'immagine ma le caratteristiche geometriche degli oggetti – in questo caso triangoli, ellissi, segmenti di vario tipo, sfondi, testi. È una modalità comodissima per fabbricare un diagramma del genere. Una volta costruito un triangolo, con le sue frecce, il suo sfondo e i testi, per fare gli altri, basta raggruppare tutti gli elementi con una semplice riquadratura e duplicare il tutto, spostare la copia nella posizione desiderata e poi intervenire sui singoli elementi che debbano essere eventualmente cambiati, per esempio i testi in questo caso. Ciò è reso possibile dal fatto che il sistema tiene traccia di ogni singolo oggetto e di ogni sua caratteristica, in un modo che rende molto semplice manipolarlo, riprodurlo e nuovamente alterarlo in ogni successiva fase del lavoro:

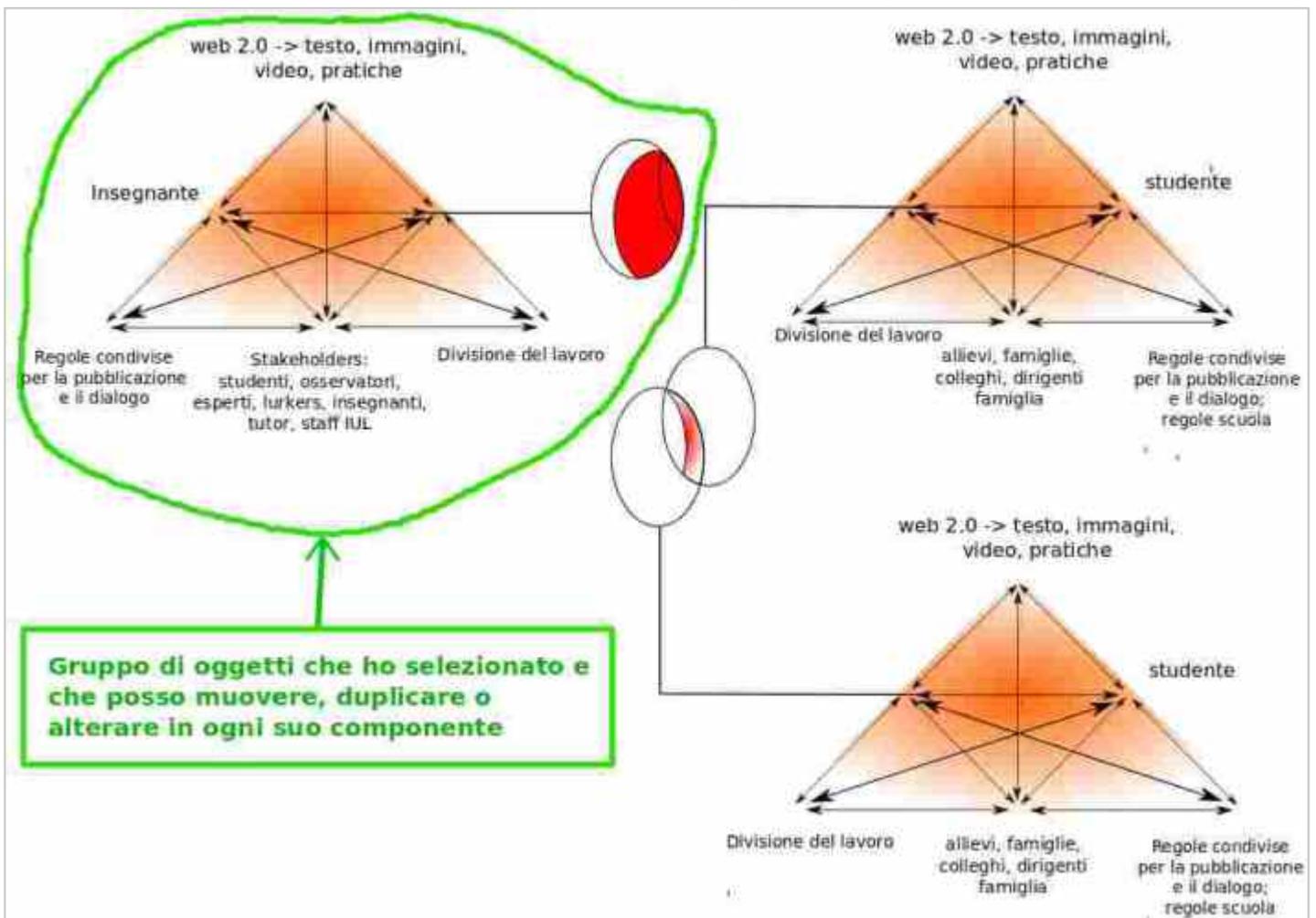


(<http://iamarf.files.wordpress.com/2014/02/activity-system-5-moving.png>)

Esempio di come, in grafica vettoriale, sia facile selezionare un gruppo di oggetti, per poi spostarlo, replicarlo o modificare in ogni suo componente. Cliccare l'immagine per vederla meglio.

Fra i casi estremi che abbiamo mostrato esiste tutta una zona intermedia di risultati che possono essere raggiunti con ambedue i sistemi. Per averne un'idea può essere interessante andare a vedere in DeviantART - (<http://www.deviantart.com/>) un social network dedicato alle creazioni grafiche – i lavori fatti sia dagli appassionati di Gimp (<http://the-gimp.deviantart.com/gallery/?2589740>) che da quelli di Inkscape (<http://inkscape.deviantart.com/favourites/>).

Anche Inkscape ha un formato di riferimento che conserva tutto ciò che serve per conservare il lavoro fra sessioni successive, è lo **SVG**. Valgono le stesse considerazioni fatte a proposito del formato XCF di Gimp. La differenza con Gimp è che Inkscape non offre formati che comprimono in modo non conservativo – per esempio si può salvare in PNG ma non in JPEG. Del resto, se vogliamo, il formato vettoriale opera intrinsecamente una forma di compressione, memorizzando solo i parametri geometrici degli elementi presenti anziché i contenuti dei pixel, che sono quasi sempre centinaia di migliaia o milioni. Ad esempio, l'immagine precedente, che è anche abbastanza complessa, è composta da 780'000 pixel ma il file in formato SVG occupa solo 85 KB. Se poi uno volesse comprimere ulteriormente un'immagine prodotta in grafica vettoriale usando il formato JPEG, per qualche giustificato motivo, non ha che da prendere la versione in PNG, caricarla in Gimp e da lì salvarla in JPEG, con il grado di compressione desiderato. Lo faccio, esagerando, per far vedere ancora il tipo di degenerazione che può venir fuori. Il file in JPEG è diventato di 29 KB ma...



(<http://iamarf.files.wordpress.com/2014/02/activity-system-5-moving.jpg>)

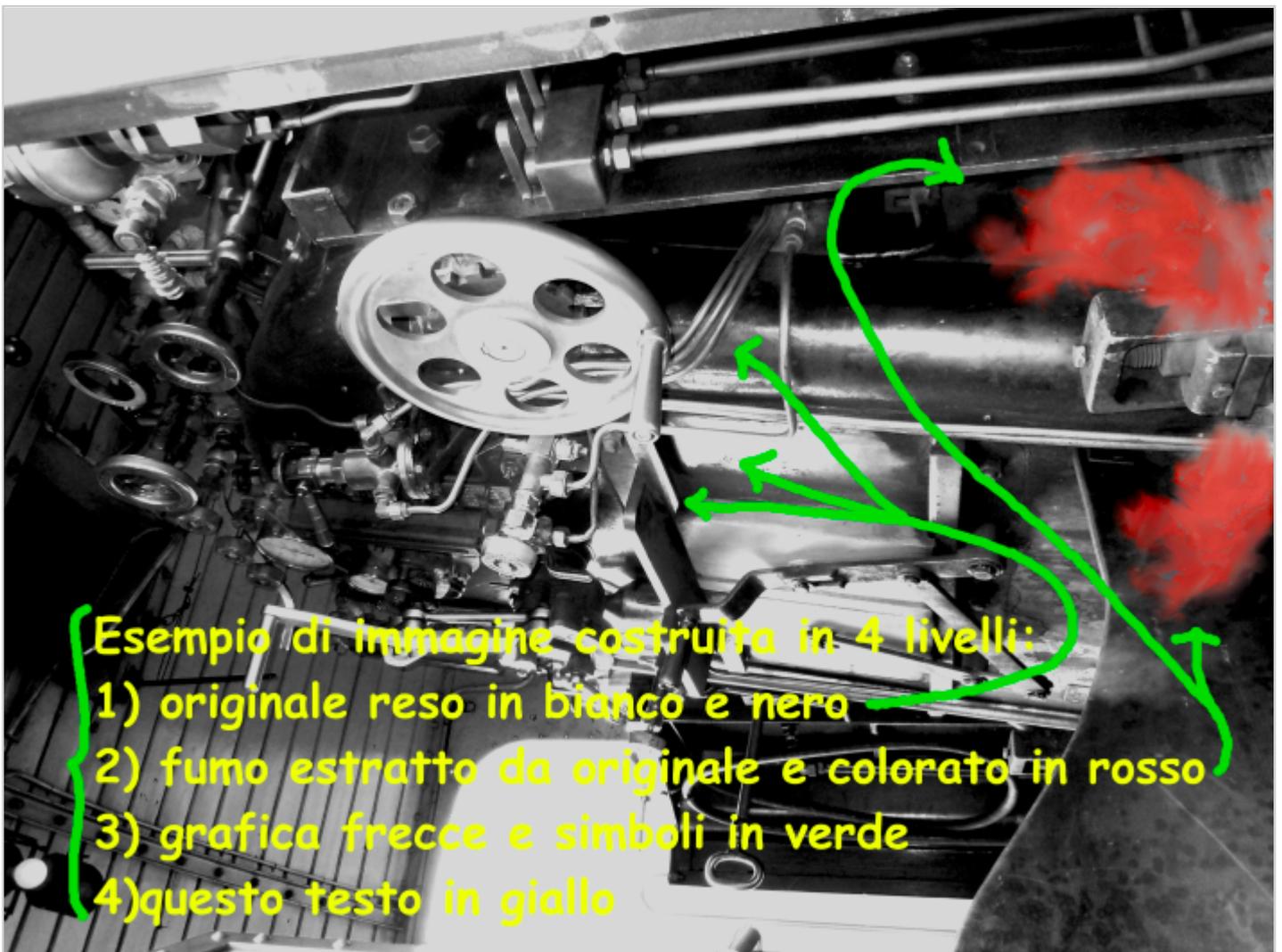
Esempio del modo in cui si deteriora un'immagine prodotta in grafica vettoriale applicandole un'eccessiva compressione JPEG. Qui l'immagine è stata prodotta lavorando con Inkscape sul suo formato nativo SVG, poi è stata esportata nel formato tipo bitmap PNG, che comprime in modo conservativo, quindi è stata caricata in Gimp per essere infine salvata in formato JPEG con qualità del 10%. Clicca l'immagine per vedere bene gli effetti.

Terzo fatto: usare i livelli (*layers*)

Infine il terzo fatto che fa la differenza in materia di elaborazione di immagini: l'esistenza dei livelli (*layers*). I livelli consentono di costruire un'immagine attraverso la sovrapposizione di piani contenenti elementi diversi di quella che sarà l'immagine risultante. Un caso tipico può essere quello della sovrapposizione di scritte ad un'immagine fotografica: la fotografia sta in un piano e i testi da sovrapporre stanno in altri piani.

Quello dei livelli è uno strumento potentissimo. Un vantaggio fondamentale è la facilità con cui si può intervenire in tempi successivi solo su alcune parti delle immagini. Cambiare un testo ormai integrato in una fotografia può essere un incubo. Cambiarlo nel suo piano, lasciando tutto il resto inalterato, è un gioco da ragazzi.

Vediamo un esempio costruito sulla solita immagine della locomotiva.



(<http://iamarf.files.wordpress.com/2014/02/dampfloko-1.png>)

Questa immagine è stata costruita con Gimp in 4 livelli: il primo contiene la fotografia resa monocromatica, il secondo contiene solo il fumo che spiffera dai meccanismi, reso in rosso, il terzo contiene gli elementi grafici in verde e il quarto i testi in giallo.

Le possibilità offerte dai livelli sono sterminate. Per esempio, sempre con Gimp, non è difficile ottenere dalla precedente un'immagine animata in GIF:

Anche il programma in grafica vettoriale Inkscape offre l'elaborazione in livelli. Tuttavia, siccome nella grafica vettoriale ogni oggetto rimane indipendente e può sempre essere riacciuffato

successivamente, da solo o in gruppo, i livelli giocano un ruolo un po' minore, anche se in certe circostanze possono tornare molto utili. Nella grafica bitmap invece l'esistenza dei livelli cambia la vita a chiunque voglia fare qualcosa di appena più elaborato di un semplice ritocco.

Coda

Non ho dato istruzioni per l'uso – come si fa a fare questo o quello. Non credo che si impari per istruzioni preconfezionate ma per suggestioni. Se la suggestione per qualcuno c'è stata allora costui proverà a fare qualcosa. Se avrà problemi lo scriverà, e noi lo aiuteremo.

Note

1. B: byte = 8 bit; 1 KB = 1024 B; 1 MB = 1024 KB; quindi 1 MB = 1024 x 1024 B, approssimativamente 1 MB è qualcosa più di un milione di B.
2. Per chi ha poca dimestichezza con il linguaggio matematico. L'equazione dice molto semplicemente che il risultato dell'operazione a sinistra dell'*uguale*, deve essere uguale a ciò che c'è alla sua destra, vale a dire uguale a 0. L'operazione a sinistra non è altro che una somma algebrica (vale a dire che c'è anche roba col segno meno) di alcuni termini, che sono a loro volta prodotto di alcuni fattori: x^2 vuol dire x moltiplicato x , $2\alpha x$ vuol dire 2 moltiplicato α moltiplicato x , eccetera. α , β e r sono le cose fisse, i cosiddetti *parametri*. Sono fisse nel senso che se io attribuisco a ciascuna di esse un valore, allora ho definito una precisa circonferenza, all'interno del mondo di tutte le infinite possibili circonferenze possibili. Rimangono x e y . Queste sono le *variabili*, quelle che servono a disegnare la circonferenza su un pezzo di carta o sullo schermo del computer, in questo caso. In sintesi, il software considera x e y come le coordinate dei pixel sullo schermo, esattamente come nella battaglia navale o in una carta geografica. Poi si "diverte" a "provare" vari valori della x , risolvendo l'equazione, cioè trovando la y in modo che valga l'uguale. Trovata così la y per ogni x , usa tali valori per individuare il pixel di tali coordinate e lo riempie di colore. Così viene fuori la circonferenza. Questa è una descrizione un po' semplificata ma è corretta.
3. Interno cabina della locomotiva a vapore HG 3/4 Nr. 1 "Furkahorn", fabbricata in Winterthur (CH) nel 1913 per operare sulla tratta a scartamento ridotto di alta montagna Brig-Furka-Disentis, successivamente venduta in Vietnam nel 1947, dove ha lavorato fino al 1993, anno nel quale è stato riportata in Svizzera per viaggiare sulla linea Oberwald-Furka-Realp, da allora operativa nella stagione estiva
4. I programmi che consentono di salvare immagini nel formato JPEG offrono sempre la possibilità di determinare la qualità regolando il compromesso compressione-qualità mediante un fattore numerico. Talvolta questo fattore è espresso come una percentuale che va da 0 a 100, ma attenzione, questo non rappresenta il fattore di compressione del file risultante ma solo un indice numerico che esprime il livello del suddetto compromesso. Programmi diversi quantificano in maniera diversa tale compromesso, che delle volte è espresso in termini differenti da quello percentuale. Quindi, quando si salva un'immagine in JPEG, dire che si è usato quel certo livello di compressione non vuol dire nulla se non si specifica anche con quale software è stato ottenuto. Ecco perché nel testo abbiamo specificato per esempio "10% secondo Gimp".
5. Il diagramma rappresenta un tentativo ingenuo del sottoscritto di descrivere l'attività dell'insegnante nel cMOOC #ltis13 sulla base dell'*activity theory* di Yrjö Engeström

(<http://www.edu.helsinki.fi/activity/people/engestro/>). Giusto per dare un'idea di cosa rappresenti il diagramma, in maniera estremamente sintetica: i triangoli sono una generalizzazione del concetto di azione mediata di Vygotsky, rappresentata con un triangolo che descrive come il soggetto agisca sull'oggetto attraverso una mediazione complessa, realizzata mediante artefatti complessi, strumenti o simboli. Tale generalizzazione prende il nome di *activity system* e tiene conto del fatto che un soggetto agisce sempre all'interno di un sistema e che la descrizione delle sue azioni rappresentano una particolare prospettiva di quella che in realtà poi risulta un'attività del sistema. Facendo riferimento al triangolo di sinistra, per esempio un insegnante opera in un sistema caratterizzato da precisi insiemi di collaboratori (in basso al centro) che condividono un certo sistema di regole (vertice basso sinistro) e operano secondo una determinata divisione del lavoro (vertice basso destro), producendo, mediante un sistema di strumenti e simboli (vertice alto), un intervento che dovrebbe intercettare le domande e le necessità degli studenti, all'interno dalle loro rispettive zone di sviluppo prossimale.

[]

[Blog at WordPress.com.](#) | [Customized Newsy Theme.](#)