

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea in Matematica

**UN NUOVO ALGORITMO
DI RIMAPPATURA DEI COLORI
PER UTENTI DALTONICI
DI CALCOLATORI**

Tesi di Laurea in Topologia Algebrica

Relatore:
Prof. MASSIMO FERRI

Presentata da:
FEDERICO PIERUCCI

‡ Sessione prima
Anno Accademico 2007/08

Indice

1	La percezione umana dei colori	4
1.0.1	Bastoncelli e coni	5
2	Il sistema RGB dei monitor	7
3	La visione nei daltonici	9
3.1	Distinzioni: rosso e verde	9
3.1.1	Dicromatismo	9
3.1.2	Linea di confusione	10
3.1.3	Dicromatismo: il blu	11
3.1.4	Acromatismo	11
3.1.5	Daltonismo debole	11
4	Cos'è già disponibile in commercio e in rete	12
5	Un nuovo punto di vista	13
5.1	Il nuovo algoritmo	13
5.1.1	Scopo	13
5.1.2	L'algoritmo e il sorgente	14
5.1.3	Il nocciolo: perché deve funzionare	19
5.1.4	Linee di confusione trasformate	20
5.1.5	Osservazioni, eventuali miglioramenti	24
5.1.6	Un errore trovato	25
5.2	Risultati	26
5.3	La sperimentazione	26
6	Conclusione	28
A	Figure a Colori	29
B	Sorgente per Mathematica	35

Introduzione

Non è difficile immaginare quali siano i problemi che i daltonici affrontano nella vita di tutti i giorni. Queste difficoltà si presentano in modo particolarmente accentuato nell'uso di applicativi per computer, in quanto, spesso, i programmatori danno un'informazione all'utente mediante colori (per esempio tasti rossi e verdi), senza curarsi delle possibili confusioni derivanti da un'anomalia nella percezione cromatica. Sono già in commercio programmi che permettono ad un programmatore di vedere la propria opera come la vedrebbe un daltonico, ma il software ad uso di utenti daltonici già disponibile ha un'efficacia molto limitata.

In questo lavoro di tesi ho sviluppato e sperimentato un algoritmo per deformare la tavolozza utilizzata in un'immagine, in modo tale che i colori più vivaci che vengono confusi dai daltonici di tipo più comune vengano trasformati in colori non ambigui, senza spostare l'ambiguità su altri colori. L'effetto è senz'altro artificiale, ma risulta efficace ed anche gradito. L'algoritmo realizza una funzione che manda i colori da \mathbb{R}^3 a \mathbb{R}^3 . Essa è continua quasi ovunque.

La prima parte della tesi introduce il metodo di visione dell'occhio umano, il funzionamento degli schermi per computer nella produzione di colori, e, parte fondamentale per comprendere appieno l'algoritmo, il funzionamento della visione dei daltonici.

La seconda parte della tesi descrive dettagliatamente l'algoritmo e presenta il codice sorgente realizzato in Matlab. Per motivi di stampa tutte le immagini a colori sono nell'appendice.

Infine si espone come sono state proposte immagini trasformate ad un soggetto daltonico, con risultati positivi.

Capitolo 1

La percezione umana dei colori

Prima di addentrarsi nella lettura dell'algoritmo è necessario conoscere un po' come funziona la visione umana dei colori. Da qui in poi, quando si parla di 'noi' intendiamo riferirci a persone non daltoniche che percepiscono i colori come la maggior parte degli umani. Usando il linguaggio tecnico 'noi' equivale a normovedenti.

Tutti crediamo di sapere di che colore è la nostra bicicletta, la frutta che mangiamo, il pesce (rosso) del nostro acquario, il semaforo. Forbici rosse e affilate: il colore ci sembra oggettivo come l'affilatura delle forbici, che non dipende da chi le usa.

Innanzitutto i colori non esistono in natura. O meglio, esiste la luce, che dal punto di vista della fisica è un'onda elettromagnetica, e ogni oggetto assorbe e diffonde solo le onde luminose di certe lunghezza d'onda. Per esempio una fragola matura illuminata dal sole diffonde luce con lunghezza d'onda principalmente compresa tra i 650 700 nm. I colori sono invece una sensazione che si genera nel sistema visivo, e dipendono da come è fatto l'occhio che guarda. È nel nostro cervello che avviene la visione dei colori, o se si vuole essere più precisi, la creazione dei colori.

Non possiamo dire che la fragola è rossa, perché la sua colorazione dipende dal fatto che siamo di una certa specie animale e non di un'altra, dipende dalle caratteristiche del nostro occhio. L'unica cosa che oggettivamente e indipendentemente dall'osservatore si verifica è l'emissione di un fascio di onde elettromagnetiche.

I tipi di occhio presenti in ogni specie animale dotata di vista sono molto diversi tra loro: occhi più adatti alla visione notturna, altri per la visione diurna, occhi formati da una miriade di cellette visive, come nelle mosche o i nostri occhi unici che funzionano come una camera oscura. Alcune specie animali riescono a vedere anche nell'ultravioletto, come gli uccelli e le api, mentre il nostro occhio ignora completamente quella banda di luce.

Non ci soffermiamo sulla visione animale o sull'ottica geometrica, la rifrazione e l'attraversamento della luce dell'occhio umano, ma cerchiamo solo di spiegare come fa il nostro

occhio a capire di che tipo è la luce che lo raggiunge e come successivamente si genera il colore. Il meccanismo può forse essere paragonato al sapore: quando assaggiamo un cibo sentiamo se è dolce, salato, amaro, ma il suo gusto non è altro che la stimolazione di alcuni recettori presenti sulla lingua umana da parte di alcune molecole. Se tocchiamo una superficie di vetro non sentiamo sapore; parallelamente se guardiamo una luce ultravioletta non vediamo colore, neanche luce. Questo è dovuto alla sensibilità del nostro occhio alla sola luce compresa tra 400 e 700nm di lunghezza d'onda, e l'ultravioletto è al di fuori di questo intervallo.

La luce che entra nell'occhio stimola dei recettori sensibili ciascuno ad un certo intervallo di lunghezze d'onda. Questi recettori si trovano soprattutto sulla parte più sensibile della retina: la fovea.

1.0.1 Bastoncelli e coni

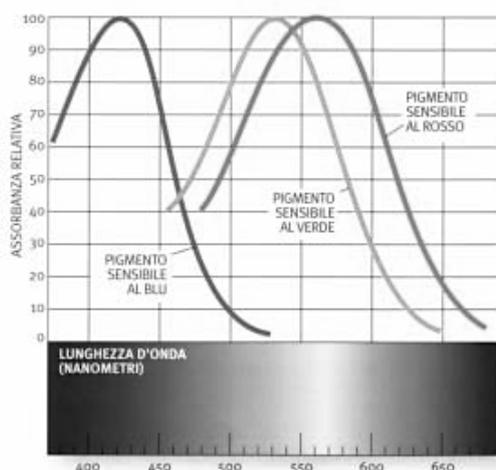


Figura 1.1: La sensibilità dei coni alla lunghezza d'onda della luce.

I bastoncelli sono i recettori più sensibili alla luce e permettono la visione notturna, ma essi non ci interessano perché la visione che generano è scotopica, cioè in bianco e nero, e per di più sono attivi solo con luce molto tenue. Infatti di notte la nostra visione di colori è pressoché nulla, le stelle ci appaiono bianche¹, un paesaggio illuminato dalla luce della luna appare grigio. Gli altri fotorecettori sono chiamati coni. Nell'occhio di un normovedente ne sono presenti 3 tipi, chiamati cono rosso, cono verde e cono blu. Ogni tipo di cono è sensibile di più alla luce di una lunghezza d'onda che alle altre: il blu con

¹Se le stelle fossero più vicine avrebbero più luminosità apparente e le vedremmo ciascuna di colore diverso a seconda del tipo di stella.

circa $420nm$, il verde $530nm$ e il rosso $560nm$ (Figura 1.1, A.1 a pag.29). In un occhio ci sono circa 6 milioni di coni.

Quando i coni sono raggiunti da una luce ciascuno di loro trasmette al cervello un segnale più forte o più debole a seconda dello stimolo ricevuto. Il cervello rielabora i segnali fino a darci la sensazione di colore.

Ci sono molti modi di organizzare i colori e di catalogarli. L'immagine A.2 a pagina 30 rappresenta solo i colori con la stessa luminosità² in un grafico, chiamato grafico cromatico CIE³. L'asse delle x rappresenta l'eccitazione del cono rosso, l'asse y del cono verde (x e y assumono valori nell'intervallo $[0, 1]$). L'eccitazione z del cono blu è $z = 1 - x - y$, perché abbiamo luminosità costante 1 (cioè $x + y + z = 1$). Quindi vediamo delle fragole rosse perché il cono rosso è molto stimolato, diversamente dai coni blu e verde. Tutti gli altri colori che vediamo in natura sono per noi semplicemente combinazioni opportune di queste 3 componenti, chiamate anche colori primari. Un limone maturo stimola molto sia il cono rosso che quello verde, perciò il cervello ci dà la sensazione di giallo che si trova esattamente a metà tra verde e rosso nel grafico dei colori. Sensazioni di bianco, nero e grigi sono corrispondenti ad una uguale stimolazione di tutti e tre i coni (al centro del grafico cromatico).

²Due colori hanno la stessa luminosità se e solo se la somma delle stimolazioni di ciascun cono è la stessa. Questo si capirà meglio quando parleremo dei colori prodotti da monitor di computer.

³La CIE è la Commissione Internazionale Illuminazione.

Capitolo 2

Il sistema RGB dei monitor

Gli schermi televisivi e i monitor dei computer sfruttano proprio il processo di scomposizione della luce nel nostro occhio. Ogni pixel dello schermo è diviso in 3 parti che producono rispettivamente luce rossa, verde e blu (da cui il nome RGB: Red, Green, Blue). Questo vale sia per gli schermi a tubo catodico dei vecchi computer e televisori, sia per gli schermi a cristalli liquidi (LCD) di ogni genere. Nel televisore queste tre parti del pixel si chiamano fosfori, e per comodità in seguito verranno denominati in questo modo anche sugli schermi LCD.

A seconda del colore che il pixel deve avere ciascun fosforo verrà illuminato molto o poco, in modo proporzionato alla stimolazione dei 3 coni corrispondente a quel colore. Grazie alle piccole dimensioni dei pixel il nostro occhio fonde i 3 fosfori e li percepisce come una unica sorgente luminosa proveniente da una sola direzione. Uno schermo giallo ha i fosfori blu spenti, i verdi e i rossi accesi, e stimola solo i nostri coni rosso e verde.

Però i fosfori che emettono la luce del pixel non riescono a produrre tutti i colori visibili da noi, ma solo un sottinsieme. Di più, le porzioni di grafico cromatico colorabili con un pixel sono diverse a seconda del tipo di schermo: più grandi, piccole, ma in ogni caso triangolare (figura A.3).

Ogni colore RGB viene rappresentato con una terna (r, g, b) , con $r, g, b \in \mathbb{N} \cap [0, 255]$. I colori del monitor possono quindi essere rappresentati con un cubo di \mathbb{R}^3 di lato 255, avente ad esempio in $(0, 0, 0)$ il nero. Infatti se in una zona dello schermo vediamo nero significa che lì non viene emessa luce. I colori diversi da bianco e nero, ma con le tre componenti uguali, o differenti l'una dall'altra di qualche unità, appaiono come grigi. Alcuni esempi sono nella tabella 2.

Il sottinsieme del cubo contenente i colori con la stessa luminosità¹ pari a 255 è proprio uno dei triangoli che vediamo nel grafico cromatico. Chiameremo questo triangolo

¹Anche per l'RGB la luminosità di (r, g, b) è definita come $I := r + g + b$.

'triangolo dei colori'. Qui la luminosità assume valori tra 0 e $765 = 3 \cdot 255$, a differenza di quanto abbiamo detto prima per l'occhio umano, in cui la luminosità assume valori tra 0 e 1. Ma è solo un fattore di scala moltiplicativo dovuto alla notazione usata.

<i>r</i>	<i>g</i>	<i>b</i>	
0	0	0	<i>nero</i>
255	255	255	<i>bianco</i>
255	0	0	<i>rosso</i>
0	255	0	<i>verde</i>
255	255	0	<i>giallo</i>
183	183	183	<i>grigio</i>
182	180	179	<i>grigiastro</i>

Tabella 2: *Alcuni colori con la loro rappresentazione in RGB. Il colore è grigiastro quando le componenti differiscono di al più di 4 o 5 unità, grigio quando le componenti sono uguali. Bianco e nero sono quindi grigi.*

Abbiamo detto che il colore $x = (255, 0, 0)$ si trova nel vertice di colore rosso del cubo, e di conseguenza nel vertice in basso a destra del triangolo, dentro il grafico cromatico. A seconda del triangolo usato, che dipende dal tipo di schermo, x è posizionato in punti diversi del grafico cromatico!

Quindi se con lo standard RGB (sRGB, in figura A.3, pagina 31), con cui abbiamo una produzione ristretta di colori, con il ProPhoto RGB possiamo coprire quasi la totalità di colori distinguibili dall'occhio umano.

Vedremo più avanti perché è bene sottolineare questa particolare differenza, se si deve implementare un programma che agisce sui colori dello schermo.

Capitolo 3

La visione nei daltonici

Il daltonismo è l'incapacità di distinguere alcuni colori che a noi appaiono diversi. Esso si presenta sotto varie forme e gradi, dipendenti dalla mancanza o alterazione dei coni dell'occhio umano. Circa l'8% dei maschi ne è affetto, mentre delle donne lo 0.5%. Alcune forme di daltonismo sono acquisite mediante eventi eccezionali quali traumi cranici, assunzione di droghe o agenti chimici in generale, glaucoma, diabete. Tuttavia di norma il difetto ha cause genetiche e non si acquisisce durante la vita, ma è presente fin dalla nascita.

3.1 Distinzioni: rosso e verde

Le forme di daltonismo conseguenti all'alterazione dei coni del rosso o del verde dipendono da un'anomalia del DNA presente sul cromosoma X. In un maschio la combinazione di cromosomi XY fa sì che quest'alterazione prevalga (2.5%). Nelle donne¹, invece, per avere la predominanza, il gene deve essere alterato in entrambi i cromosomi X, e l'alterazione deve essere dello stesso tipo (cioè per esempio cecità al verde in entrambi gli X). La percentuale di donne con daltonismo è quindi molto minore: solo lo 0.02%. La donna può essere portatrice sana, cioè può trasmettere l'alterazione ereditariamente senza che lei sia daltonica, se è anomalo solo un X.

3.1.1 Dicromatismo

Il dicromatismo è l'assenza di un tipo di cono dalla retina. Se c'è l'assenza del cono del rosso o del verde, la persona è cieca a quel colore. Que-

¹hanno coppia di cromosomi XX.

sto non significa che gli oggetti rossi (o verdi, rispettivamente) appaiono neri. I colori appaiono semplicemente privati della componente rossa, mentre gli altri due coni sono stimolati esattamente come in un normovedente, come si può intuire dal grafico A.1 (pagina 29). Il daltonico non percepisce nessuna stranezza, il mondo è per lui colorato in modo vario. Anzi, è curioso il fatto che mentre noi abbiamo tre coni, gli uccelli, i sauri, le tartarughe e molti pesci ne hanno quattro[2]. Oltre a vedere tutti i nostri colori, essi riescono a distinguere alcune tonalità che percepiamo come uguali. Per esempio un fiore giallo viene visto in modo completamente diverso da un uccello: la metà del petalo vicino al polline diffonde anche raggi ultravioletti, mentre l'altra metà no. L'effetto per gli uccelli è come se vedessimo il fiore con il bordo della corolla giallo e l'interno blu. Possiamo considerarci come uccelli daltonici perché abbiamo un cono in meno di loro. Da questo punto di vista è molto facile intuire che per un uomo daltonico i colori che lui vede gli sembrano normali. Ugualmente anche chi vede il mondo sempre in bianco e nero considera questo fatto naturale e non riesce ad immaginare cosa siano i colori.

3.1.2 Linea di confusione

Definiamo la linea di confusione come l'insieme dei colori o dei punti (nel grafico cromatico) che un daltonico vede come se fosse lo stesso colore.

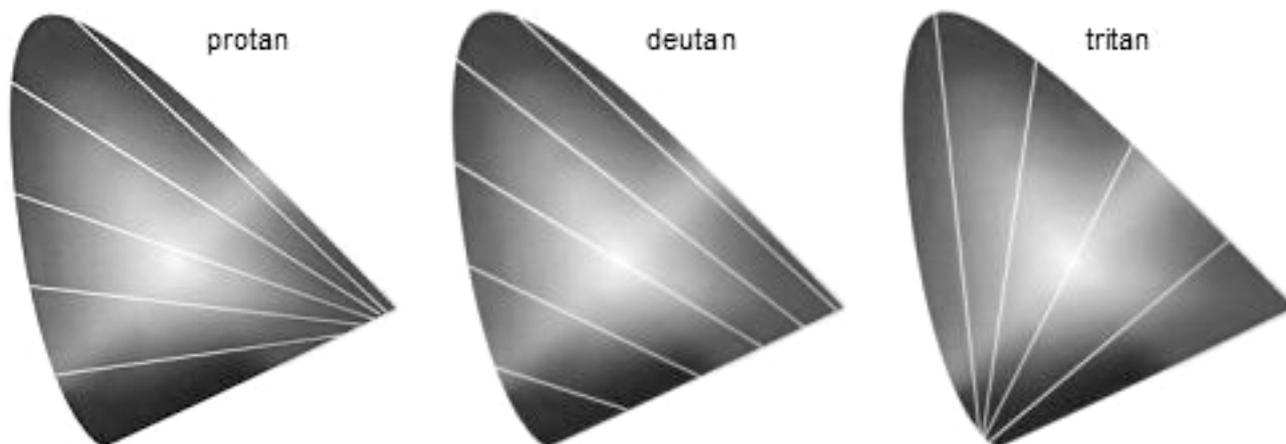


Figura 3.1: Linee di confusione di un cieco al rosso, al verde e al blu rispettivamente.

La disposizione di queste linee dipende da quale cono è assente, quindi abbiamo tre famiglie di linee di confusione. Il grafico cromatico relativo ad una persona dicromatica risulta enormemente modificato: anziché avere i colori con la stessa luminosità disposti su un piano, li troviamo disposti su un segmento. Ad ogni colore del grafico cromatico viene associato un punto del segmento: tutti i colori disposti lungo la linea di confusione

corrispondono allo stesso punto del segmento, cioè lo stesso colore per un daltonico. Osserviamo che sia i verde-ciechi che i rosso-ciechi hanno il giallo, il rosso e il verde molto vicini ad una linea di confusione: come conseguenza questi tre colori non vengono distinti, ma appaiono come unico. L'unico caso in cui dei colori sulla stessa linea di confusione siano distinguibili è quello in cui abbiano luminosità diverse.

3.1.3 Dicromatismo: il blu

Nel caso manchi il cono del blu si ha cecità al blu. Anch'esso genera delle linee di confusione, ma in questo caso vengono confusi alcuni blu con il rosso, altri blu con il verde, altri con il giallo. Questo difetto è il più raro dei tre tipi di dicromatismo, dipende da un cromosoma diverso da X e la sua diffusione è equilibrata tra uomini e donne (0.008% della popolazione).

3.1.4 Acromatismo

La mancanza di due coni genera l'acromatismo. L'acromatico non percepisce colori, ma solo luminosità. È molto semplice immaginare la loro visione: proprio come un film in bianco e nero.

Alcune persone hanno un acromatismo basato sull'assenza di tutti e tre i coni. Esse vedono attraverso i bastoncelli², molto più sensibili alla luce rispetto ai coni. Questo genera una visione in bianco e nero accecante, come descritto nel libro di Oliver Sacks[4]. L'acromatismo è molto raro.

3.1.5 Daltonismo debole

In questa forma di daltonismo sono presenti i tre coni, ma uno di essi è meno sensibile e altera la percezione dei colori. Oppure in altri casi i picchi di sensibilità del rosso e del verde, che normalmente si trovano a 560nm e 530nm, sono più vicini³. Le curve del rosso e del verde nel grafico A.1 risultano quasi sovrapposte. L'effetto è più o meno accentuato e varia da persona a persona. Si stima che le persone affette da daltonismo debole, in forma leggera o più marcata, siano il 5,5% dei maschi e lo 0,4% delle donne.

²Già introdotti a pagina 5.

³La distanza tra i picchi può arrivare addirittura a 1nm.

Capitolo 4

Cos'è già disponibile in commercio e in rete

Per migliorare la visione dei daltonici sono stati già studiati molti metodi. Ci sono occhiali con una lente di un colore e l'altra trasparente. Indossando gli occhiali alcuni colori sono visti allo stesso modo con entrambi gli occhi, altri sono alterati solo per un occhio. Così la persona ha un miglioramento nella distinzione dei colori. Questi occhiali sono prodotti da **Colorlite Spa**. [6]

Per quanto riguarda il computer, la soluzione più utilizzata è quella di creare direttamente interfacce grafiche e siti web in modo che non abbiano colori ambigui, cioè colori che i daltonici non distinguono bene. Un esempio è la '**Ruota dei colori accessibili**' [7].

Un altro metodo è quello di risolvere il problema a valle. Data un'immagine o una pagina web qualsiasi si cerca di modificarne i colori per renderla accessibile ai daltonici. Alcuni programmi permettono di modificare sul momento la luminosità di alcuni colori lasciando invariati gli altri, o di scambiare le componenti di r, g, b tra di loro. L'utente sceglie per ogni immagine come modificarla, ma non sempre si riesce a trovare la giusta combinazione, soprattutto se l'immagine ha un vasto assortimento di colori. Ad esempio in una cartina degli autobus si riesce a distinguere le linee a due a due, ma non si trova il modo di distinguerle tutte assieme.

Un programma abbastanza efficace, sebbene molto rudimentale dal punto di vista teorico, è **Eyepilot**TM [8], che risolve il problema degli autobus. L'utente clicca su una parte di immagine, ad esempio la linea 33, e tutto ciò che ha un colore diverso diventa grigio. Lasciando il click l'immagine torna come prima, colorata. Oppure si può far diventare nero il colore cliccato, in modo da evidenziarlo anche in altre parti dell'immagine.

Infine, esistono programmi capaci di simulare la visione di un daltonico, come ad esempio **Vischeck** [9]. Questo programma effettua in pratica la mappatura dal grafico cromatico al segmento dei colori dei daltonici, cioè agisce come il sistema visivo di un daltonico cieco al verde.

Capitolo 5

Un nuovo punto di vista

Fino ad ora i programmi esistenti creati per evitare l'ostacolo del daltonismo sono di ausilio alla creazione ex novo di immagini non ambigue, ma senza poter modificare una foto esistente, oppure applicano un semplice cambiamento dell'immagine che migliora la distinzione di alcuni colori, a scapito di altri. Di fatto l'immagine finale ha gli stessi problemi di quella iniziale, l'ambiguità è presente comunque tra alcuni colori. È un po' come passare da un tipo di dicromatismo all'altro.

5.1 Il nuovo algoritmo

5.1.1 Scopo

Si vuole creare un algoritmo che trasforma un'immagine qualsiasi. Vogliamo che l'immagine finale conservi tutte le informazioni riguardanti i colori, e che il daltonico sia capace di distinguere tutti i colori senza il bisogno dell'immagine iniziale. Si vuole cioè eliminare il problema dell'ambiguità da tutta l'immagine, contemporaneamente per tutti i colori.

Questo è teoricamente impossibile, per il fatto che piano e segmento hanno dimensione diversa. Però possiamo trasformare l'immagine in modo che tutti i colori sul bordo del grafico cromatico, o meglio, del triangolo dell'RGB, siano distinguibili. Questo è possibile teoricamente perché il bordo e il segmento hanno entrambi dimensione 2 come sottinsiemi di \mathbb{R}^2 . Abbiamo cercato quindi un'applicazione continua con dominio il triangolo dei colori e codominio metà del triangolo. I colori del bordo sono identici a due a due per un daltonico, perciò vengono sostituiti in modo da utilizzare solo colori di metà bordo. Così l'immagine finale ha meno colori, ma se due colori del semi-bordo sono distinti per noi, allora lo sono anche per i daltonici. Per la continuità della funzione questo è vero anche per i colori vicino il bordo. Col nostro algoritmo i colori centrali, cioè i grigi, rischiano di restare confusi con qualche colore. Ma a noi interessano i colori più vivaci: vogliamo

che quando la luce colpisce i coni essi siano stimolati in modo diverso da ciascun colore vivace. Analizziamo ora il procedimento.

5.1.2 L'algoritmo e il sorgente

Va premesso che il cambiamento di colore va fatto pixel per pixel, indipendentemente dalla sua posizione nell'immagine e dai colori dei pixel vicini, ma dipende solo dal colore del pixel stesso. L'immagine è considerata come una matrice con elementi in \mathbb{R}^3 . Il software utilizzato per implementare questo algoritmo è Matlab[©], versione 5.3 (R11) 15-1-1999. Sono necessari i Toolboxes: 'Symbolic Math', 'Image Processing'.

Inizio

Sia p un pixel dell'immagine,
 $c(p) = (\alpha, \beta, \gamma)$ il suo colore in RGB,
 $\alpha, \beta, \gamma \in \mathbb{N} \cap [0, 255]$,
sia $I(p) := \alpha + \beta + \gamma$ la luminosità di p .

Fissiamo p e chiamiamo per comodità $c := c(p)$, $I := I(p)$.
Nel caso in cui $\alpha = \beta = \gamma$, l'algoritmo non modifica c , che è un grigio. Quindi da qui in poi consideriamo $c \neq (k, k, k)$, $k \in \mathbb{R}$.

Ecco l'inizio del programma in Matlab[©]. Viene caricata l'immagine e scomposta nelle tre componenti di colore. Attenzione: la notazione è diversa da quella con cui si spiega l'algoritmo, e le variabili sono vettori o matrici, in double, come di consueto in Matlab[©].

```
t0 = fix(clock); %per misurare il tempo di calcolo
rd3=sqrt(3); %radice quadrata di 3 (aumenta la velocita')

b=imread('lego2','JPEG');
a=double(b);
s=size(a); % trova la dimensione della matrice del colore 'a'

%divisione nelle 3 componenti di colore
R= a(1:s(1),1:s(2),1); %rosso
G= a(1:s(1),1:s(2),2); %verde
B= a(1:s(1),1:s(2),3); %blu

I=R+G+B+eps; % la luminosita'
```

Nota: eps è il più piccolo numero double positivo, viene aggiunto per evitare di trattare a parte i casi in cui si rischia di dividere per zero.

Parte 1

Il primo passo consiste nel proiettare il colore c dal cubo RGB al triangolo dei colori. Troviamo $c_0 := (\alpha_0, \beta_0, \gamma_0) := \frac{c \cdot 255}{I} = \left(\frac{\alpha}{I}, \frac{\beta}{I}, \frac{\gamma}{I}\right)$. c_0 appartiene al piano affine $x + y + z = 255$, perché sommando le componenti si ha $\alpha_0 + \beta_0 + \gamma_0 = 255$.

Il colore c_0 è proprio la proiezione del generico colore c sul triangolo dei colori¹.

Ora prendiamo $c_1 := \frac{c_0}{255}$ che si trova sul triangolo

$$T = \{(x, y, z) \in \mathbb{R}^3 \mid x + y + z = 1, x, y, z \geq 0\}$$

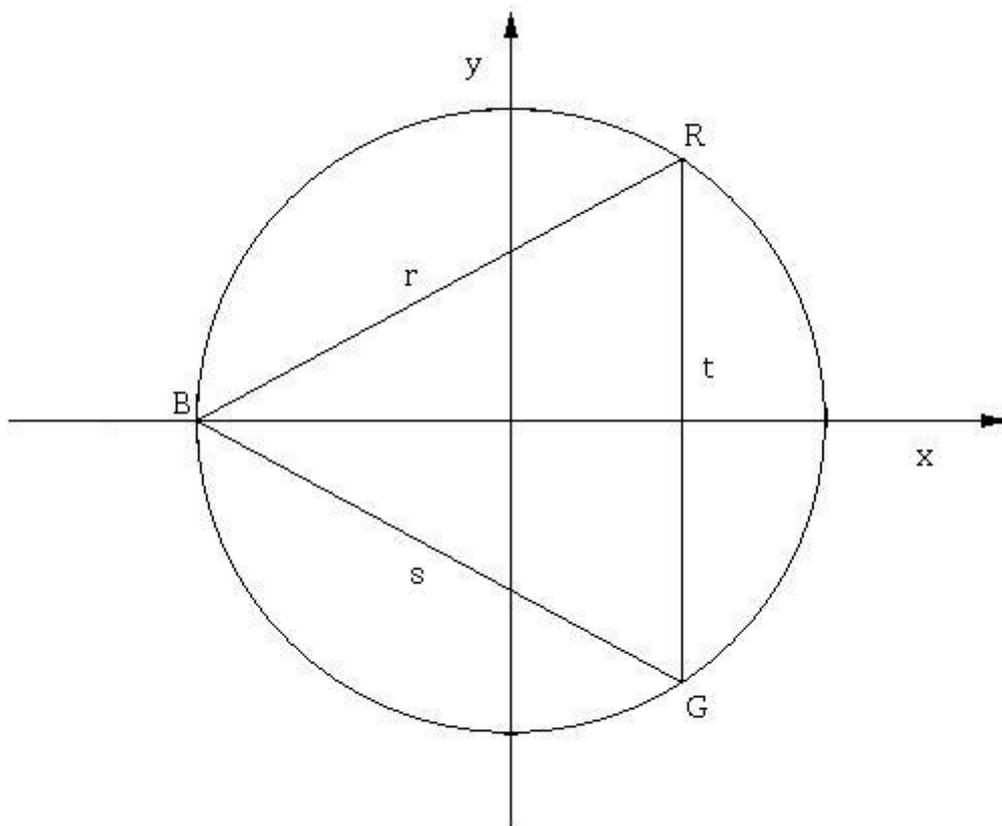


Figura 5.1: T è il triangolo di vertici B, G, R

Vogliamo un sistema di riferimento per il piano di T , con un cambiamento definito

¹Per definizione. Si veda il capitolo 2.

in questo modo:

$$\begin{aligned}
 T &\longmapsto \mathbb{R}^2 \\
 (1, 0, 0) &\longrightarrow R := \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right) \\
 (0, 1, 0) &\longrightarrow G := \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right) \\
 (0, 0, 1) &\longrightarrow B := (-1, 0) \quad .
 \end{aligned}$$

Troviamo tre rette $r, s, t \subset \mathbb{R}^2$:

$$\begin{aligned}
 r &= \{(x, y) \in \mathbb{R}^2 \mid x - \sqrt{3}y + 1 = 0\} \text{ , passante per } B \text{ e } R \\
 s &= \{(x, y) \in \mathbb{R}^2 \mid x + \sqrt{3}y + 1 = 0\} \text{ , passante per } G \text{ e } B \\
 t &= \{(x, y) \in \mathbb{R}^2 \mid x - \frac{1}{2} = 0\} \text{ , passante per } R \text{ e } G.
 \end{aligned}$$

Il triangolo che formano è equilatero e inscritto nella circonferenza S^1 . Possiamo rappresentare R, G, B come numeri complessi:

$$\begin{aligned}
 R &= \frac{1}{2} + \frac{\sqrt{3}}{2}i; \\
 G &= \frac{1}{2} - \frac{\sqrt{3}}{2}i; \\
 B &= -1.
 \end{aligned}$$

Si ottiene che $c_1 = \alpha R + \beta G + \gamma B = \alpha \left(\frac{1}{2} + \frac{\sqrt{3}}{2}i\right) + \beta \left(\frac{\alpha + \beta - 2\gamma}{2}\right) + \frac{\sqrt{3}}{2}(\alpha - \beta)i$

$$\implies \text{le coordinate di } c_1 \text{ in } \mathbb{R}^2 \text{ sono: } \begin{cases} x = \frac{\alpha + \beta - 2\gamma}{2} \\ y = \frac{\sqrt{3}}{2}(\alpha - \beta) \end{cases}$$

Ora vogliamo esprimere c_1 in coordinate polari: $c_1 = \rho e^{i\theta}$, $\rho \in (0, \infty)$, $\theta \in [0, 2\pi)$.²

$$\begin{cases} \theta = \begin{cases} \frac{\pi}{2} - \arctan\left(\frac{x}{y}\right) & \text{per } y > 0 \\ \frac{3\pi}{2} - \arctan\left(\frac{x}{y}\right) & \text{per } y < 0 \\ 0 & \text{per } y = 0, x = 1 \\ \pi & \text{per } y = 0, x = -1 \end{cases} \\ \rho = \sqrt{x^2 + y^2} \end{cases} .$$

²Per quanto detto prima sappiamo che $\rho \in (0, 1]$.

θ , in particolare, è una determinazione dell'argomento su $\mathbb{R}^2 \setminus \{(x, y) \in \mathbb{R}^2 \mid y = 0, x \geq 0\}$. Ecco il testo.³

```
%proiezione sul triangolo
R=R./I;
G=G./I;
B=B./I;

m=(R+G-2*B)./2;          %valori nel piano (m=x, n=y)
n=sqrt(3)*(R-G)./2;

r=sqrt(m.^2+n.^2); %trasformazione in coordinate polari
t=arg(m,n);
```

Parte 2

Adesso abbiamo una notazione efficace per lavorare con i colori. Possiamo sfruttare la loro rappresentazione nel piano complesso per effettuare la rimappatura. La prima cosa da fare è mandare i punti del triangolo sulla circonferenza circoscritta, e i punti dell'interno del triangolo nell'interno del cerchio. Questa applicazione deve essere un omeomorfismo, perché vogliamo che non ci siano discontinuità nello spostamento dei punti. Più avanti ci servirà anche la sua inversa, anch'essa continua. La funzione dell'espansione nel cerchio utilizzata nel nostro algoritmo è:

$$\left\{ \begin{array}{l} \theta_2 = \theta \\ \rho_2 = \left\{ \begin{array}{ll} \frac{\rho}{\sqrt{x^2+y^2}} & , \theta \in [0, \frac{\pi}{3}] \cup [\frac{5\pi}{3}, 2\pi) \\ \frac{|2x|}{\rho} & , \theta \in (\frac{\pi}{3}, \pi] \\ \frac{\sqrt{y^2+x^2}}{|\sqrt[3]{y-x}|} & , \theta \in (\pi, \frac{5\pi}{3}) \\ \frac{\sqrt{y^2+x^2}}{|\sqrt[3]{y+x}|} & \end{array} \right. \end{array} \right. ,$$

³La funzione $arg()$ non è presente nella libreria di Matlab[©]. Per chi volesse far funzionare il programma ecco il codice della funzione da inserire in un m-file di nome arg :

```
function t =arg(x,y) %determinazione dell'argomento
s=size(y);
t=zeros(s(1),s(2));
for j=1:s(1)*s(2)
    t(j)=atan2(y(j),x(j));
    if t(j)<0
        t(j)=t(j)+2*pi;
    end
end
end
```

in cui il punto iniziale c_1 viene rappresentato sia con le coordinate cartesiane che con le coordinate polari. Ma questa è solo una notazione per rendere più esplicito il significato geometrico della funzione.

Ora spieghiamo perché va bene questa funzione.

Chiamiamo u la retta di \mathbb{R}^2 passante per $(0, 0)$ e c_1 .

Definiamo

$$P := \begin{cases} t \cap u & \text{per } \theta \in [0, \frac{\pi}{3}] \cup [\frac{5\pi}{3}, 2\pi) \\ r \cap u & \text{per } \theta \in (\frac{\pi}{3}, \pi) \\ s \cap u & \text{per } \theta \in [\pi, \frac{5\pi}{3}) \end{cases}. \quad (5.1)$$

Calcoliamo ora $P = (X, Y) \in \mathbb{R}^2$ in funzione di $c_1 = (x, y)$. Intersechiamo le rette come descritto nella 5.1 e risolviamo i 3 sistemi di equazioni.

$$\bullet (t \cap u) \quad \begin{cases} X = \frac{1}{2} \\ Y = \frac{y}{x}X \end{cases} \Rightarrow \begin{cases} X = \frac{1}{2} \\ Y = \frac{y}{2x} \end{cases} \Rightarrow |P| = \frac{\sqrt{x^2 + y^2}}{|2x|} =: d_2$$

$$\bullet (r \cap u) \quad \begin{cases} X - \sqrt{3}Y + 1 = 0 \\ Y = \frac{y}{x}X \end{cases} \Rightarrow \begin{cases} X - \frac{\sqrt{3}y}{x}X = -1 \\ Y = \frac{y}{x}X \end{cases} \Rightarrow \begin{cases} X = \frac{1}{\sqrt{3}\frac{y}{x}-1} = \frac{x}{\sqrt{3}y-x} \\ Y = \frac{y}{\sqrt{3}y-x} \end{cases} \\ \Rightarrow |P| = \frac{\sqrt{x^2 + y^2}}{|\sqrt{3}y-x|} =: d_1$$

$$\bullet (s \cap u) \quad \begin{cases} X + \sqrt{3}Y + 1 = 0 \\ Y = \frac{y}{x}X \end{cases} \Rightarrow \begin{cases} X + \frac{\sqrt{3}y}{x}X = -1 \\ Y = \frac{y}{x}X \end{cases} \Rightarrow \begin{cases} X = \frac{-1}{\sqrt{3}\frac{y}{x}+1} = -\frac{x}{\sqrt{3}y+x} \\ Y = -\frac{y}{\sqrt{3}y+x} \end{cases} \\ \Rightarrow |P| = \frac{\sqrt{x^2 + y^2}}{|\sqrt{3}y+x|} =: d_3$$

I valori d_1, d_2, d_3 sono ottenuti intersecando u rispettivamente con t, r, s , a seconda di dove si trova c_1 , in particolare $|P| = d_1$ ovvero $|P| = d_2$ ovvero $|P| = d_3$.

Per qualsiasi punto c_1 del triangolo si ha $\rho \leq |P|$, perché P è sul perimetro. Quindi per mandare i punti del triangolo nel cerchio basta lasciare invariato l'angolo θ e dividere ρ per $|P|$.

%espansione radiale

a1=pi/3; %a1, a2, a3 vengono calcolati prima per velocizzare

a2=a1*5;

a3=a1*6;

```

for j=1:(s(1)*s(2))
    if t(j)<a1 | a2<t(j)
        r(j)=r(j)*sqrt(4*m(j)^2/(m(j)^2+n(j)^2+eps));
    else if n>0
        r(j)=r(j)*abs(rd3*n(j)-m(j))/(eps+sqrt(m(j)^2+n(j)^2));
    else r(j)=r(j)*abs(rd3*n(j)+m(j))/(eps+sqrt(m(j)^2+n(j)^2));
    end
end
end

```

5.1.3 Il nocciolo: perché deve funzionare

Parte 3

Questa parte è semplicissima dal punto di vista pratico. Bisogna però spiegare bene il motivo teorico per cui questa trasformazione soddisfa i nostri scopi.

La funzione è:

$$\begin{cases} \theta_3 = \frac{\theta_2}{2} \\ \rho_3 = \rho_2 \end{cases} .$$

%dimezzamento

t=t/2;

Abbiamo detto che vogliamo migliorare la visione dei dicromatici ciechi al rosso o al verde. Ora concentriamoci sui primi due grafici di figura 3.1 a pagina 10, e immaginiamo di disegnarci dentro il triangolo dei colori 'sRGB'⁴ nella posizione in cui lo vediamo nella figura di pagina 31.

In $\theta_2 = 0$ si trova il colore giallo. Poi aumentando θ_2 si passa al rosso, poi al viola, blu con $\theta = \pi$, ciano, verde e di nuovo al giallo per $\theta \rightarrow 2\pi$. Ogni⁵ linea di confusione che interseca il perimetro del triangolo ha due punti di intersezione col triangolo. Come spiegato nel capitolo 3, i dicromatici vedono uguali questi due colori. Dimezzando l'angolo il giallo resta tale, il rosso diventa un poco più arancione, i viola più rossastri e così via fino al verde che diventa violaceo e il verde-giallognolo che diventa blu. Il triangolo dei colori finale non contiene più il verde, o meglio, il verde e tutti i colori su metà del triangolo spariscono dalla foto che volevamo modificare. I colori rimanenti nel perimetro del triangolo non vengono più accoppiati a due a due dalle linee di confusione, perché ora si trovano tutti solo su metà del perimetro. Ci potrebbero essere problemi con l'arancione, ma per ora non ne teniamo conto.

⁴Standard RGB, quello più comune, ma il ragionamento è valido anche se si usa uno degli altri triangoli.

⁵Tranne le linee due per due vertici del triangolo.

5.1.4 Linee di confusione trasformate

Ora proviamo a vedere come vengono modificate le linee di confusione. Ho realizzato questi grafici con Mathematica⁶. Il codice sorgente da inserire in Mathematica è riportato a pagina 35.

Questi grafici rappresentano passo-passo come le 4 parti dell'algoritmo modificano le linee di confusione. Si procede utilizzando la retroimmagine della funzione di rimappatura, o meglio, proprio la sua funzione inversa, visto che se si esclude una semiretta la funzione è un omeomorfismo.

La prima figura (5.2) mostra la metà del triangolo dei colori, quelli presenti sull'immagine rimappata. Le linee di confusione sono quelle associate al daltonismo, già introdotte a pagina 10.

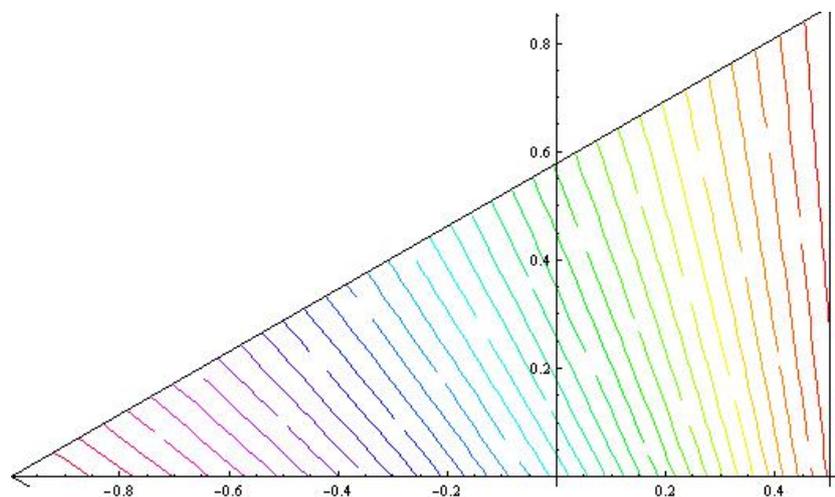


Figura 5.2:

Ora applichiamo la funzione di espansione nel semicerchio, ottenendo le linee della figura 5.3.

Adesso anziché dimezzare raddoppiamo l'angolo, perché stiamo applicando la funzione inversa (5.4).

Infine riportiamo nel triangolo dei colori. Il significato di questa figura è il medesimo di quella iniziale con le linee di confusione rettilinee. Il triangolo è il triangolo dei colori e tutti i colori attraversati dalla medesima linea appaiono ad un daltonico indistinguibili. Non confondiamo i colori con cui sono state disegnate le linee con quelli del triangolo⁷. In questo disegno (5.5) i colori servono solo per distinguere una linea dall'altra. Così

⁶Versione 6.0, prodotto dalla Wolfram.

⁷Cioè i colori presenti nel grafico cromatico.

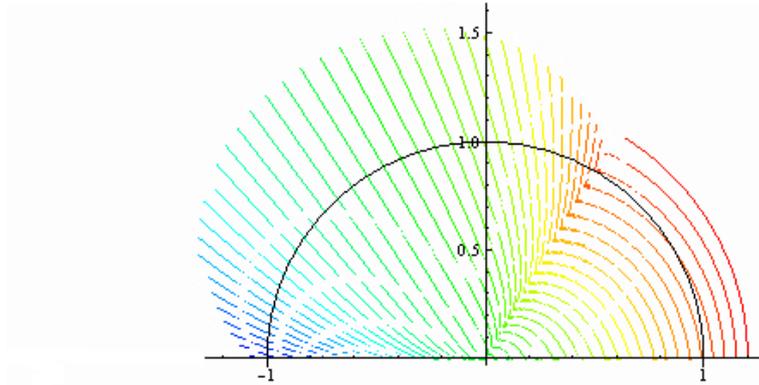


Figura 5.3:

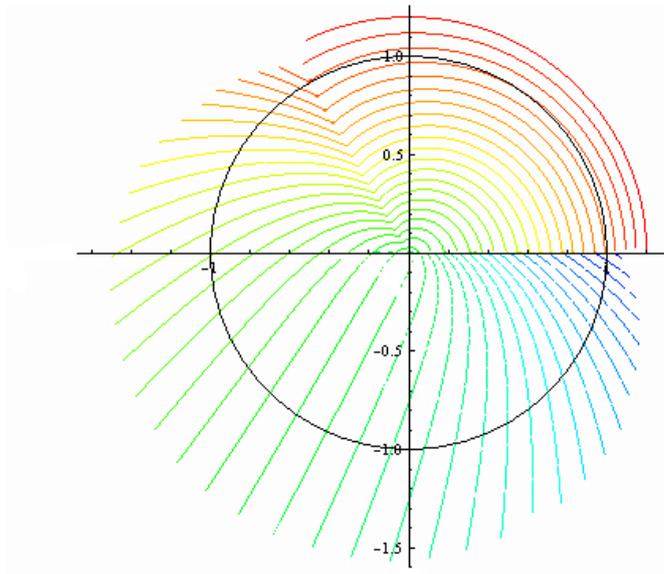


Figura 5.4:

i colori che prima venivano confusi sono stati scambiati con altri ben distinguibili. I colori che prima erano distinguibili lo restano. Se effettivamente si verifica quello che abbiamo previsto per via teorica, allora abbiamo raggiunto il nostro scopo. Parleremo

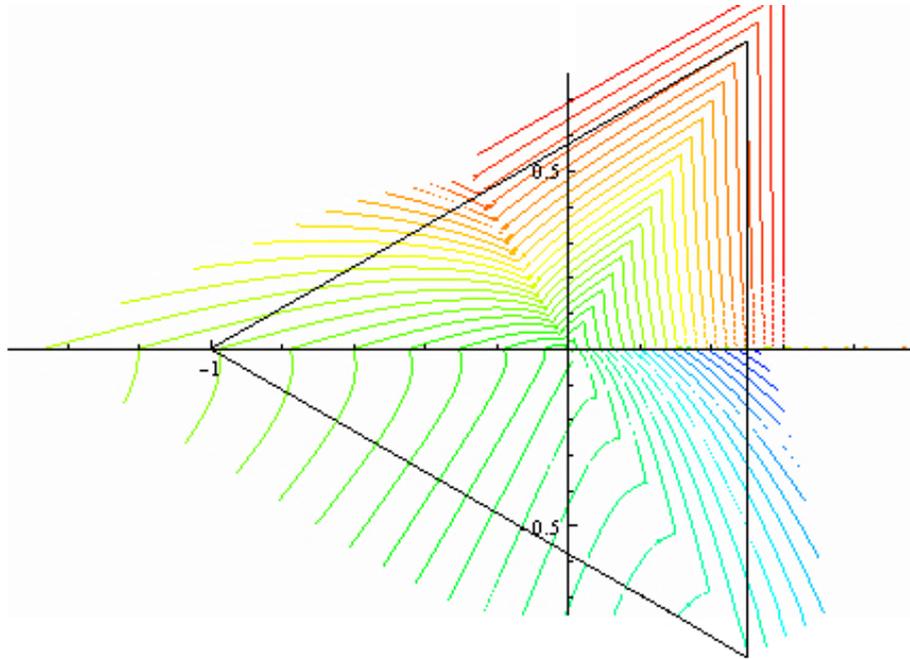


Figura 5.5: *Nuove linee di confusione nel triangolo dei colori per le immagini rimappate: la colorazione delle righe indica che le parti del grafico toccate dalla stesa riga sono viste dello stesso colore.*

della sperimentazione e di eventuali miglioramenti più avanti.

Parte 4

La parte finale dell'algoritmo comprende le funzioni inverse delle precedenti.

Trasformiamo in coordinate cartesiane:
$$\begin{cases} x_3 = \rho_3 \cos \theta_3 \\ y_3 = \rho_3 \sin \theta_3 \end{cases}$$

Poi applichiamo l'inversa dell'espansione.

$$\begin{cases} \theta_4 = \theta_3 \\ \rho_4 = \begin{cases} \rho_3 \sqrt{\frac{x_3^2 + y_3^2}{4x_3^2}} & , \theta_3 \in [0, \frac{\pi}{3}] \\ \rho_3 \frac{\sqrt{y_3^2 + x_3^2}}{|\sqrt[3]{y_3} - x_3|} & , \theta_3 \in (\frac{\pi}{3}, \pi) \end{cases} \end{cases} ,$$

e trasformiamo in coordinate cartesiane
$$\begin{cases} x_4 = \rho_4 \cos \theta_4 \\ y_4 = \rho_4 \sin \theta_4 \end{cases} .$$

`%espansione nel (semi)triangolo`

`for j=1:(s(1)*s(2))`

`if t(j)<a1`

`r(j)=r(j)*sqrt((m(j)^2+n(j)^2)/(eps+4*m(j)^2));`

`else r(j)=r(j)*(sqrt(m(j)^2+n(j)^2)/(eps+abs(rd3*n(j)-m(j))));`

`end`

`end`

`m=r.*cos(t); %ricalcolo le coordinate`

`n=r.*sin(t);`

Infine immergiamo il triangolo in \mathbb{R}^3 , facendo in modo che $\alpha_4 + \beta_4 + \gamma_4 = 1$:

$$\begin{cases} \alpha_4 = \frac{1}{3}(1 + x_4 + \sqrt{3}y_4) \\ \beta_4 = \frac{1}{3}(1 + x_4 - \sqrt{3}y_4) \\ \gamma_4 = \frac{1}{3}(1 - 2x_4) \end{cases} . \quad (5.2)$$

`%denormalizzo`

`n=rd3.*n; % (velocizza)`

`R=(m+1+n)/3;`

`G=(m+1-n)/3;`

`B=(1-2*m)/3;`

Adesso basta moltiplicare per I e ogni colore si sposta all'interno del cubo per ottenere la luminosità giusta.

$$\begin{cases} \alpha_5 = I\alpha_4 \\ \beta_5 = I\beta_4 \\ \gamma_5 = I\gamma_4 \end{cases} \quad (5.3)$$

```
a(1:s(1),1:s(2),1)=R.*I; % ricostruisce la matrice 3-dimensionale
a(1:s(1),1:s(2),2)=G.*I;
a(1:s(1),1:s(2),3)=B.*I;
```

Alla fine il programma scrive l'immagine ottenuta su un file.

```
a=uint8(a); %trasforma i numeri in unsigned int, richiesti per le immagini

imwrite(a,'immagine trasformata.jpg','jpeg'); %scrittura su file dell'immagine
c=[b a];
image(c) %fa apparire l'immagine in una finestra sullo schermo
'tempo impiegato (secondi):'
etime(fix(clock),t0) %calcola il tempo impiegato
'completato!'
```

Il tempo impiegato per trasformare immagini di 200 KB, come quella a pagina 32, è di circa 7 secondi per un computer con processore 1.83 GHz Intel Core 2 Duo, ram 1 GB 667MHz.

5.1.5 Osservazioni, eventuali miglioramenti

Può sembrare strano, o perlomeno fa venire qualche perplessità, il fatto che sia stata scelta una funzione lineare così semplice ($\theta_3 = \frac{\theta_2}{2}$).

In realtà questo algoritmo è solo un punto di partenza per migliorare la visione. Ci interessava solo provare che questa strada è effettivamente percorribile e che migliorando la funzione che modifica θ_2 si potrebbero ottenere risultati veramente efficaci.

Avevamo detto che alcuni gialli e arancioni restano confusi. Questo si verifica perché essi sono gli unici colori che cambiando la loro posizione restano nella stessa linea di confusione. Si potrebbe migliorare mandando θ_2 in $[\alpha_0, \pi]$, $\alpha_0 \in (0, \frac{\pi}{2})$, anziché in $[0, \pi]$. Abbiamo già iniziato ad implementare modifiche, ma non avevamo buoni criteri per stabilire quale è il risultato migliore. Per risultato migliore si intende una trasformazione che vada bene per ciechi al rosso e al verde e che si adatti ai diversi tipi di schermi per computer.

Alcuni schermi producono più colori di altri, quindi la posizione del triangolo dei colori

all'interno del grafico cromatico cambia. Di conseguenza, le linee di confusione vengono tracciate in modo leggermente diverso nel triangolo dei colori. Questo è ininfluenza nel nostro lavoro, ma va tenuto in considerazione se si vuole perfezionare l'algoritmo. L'ideale sarebbe crearne più versioni, ciascuna migliorata per un tipo di schermo in particolare.

Scegliamo $\alpha_0 = \frac{\pi}{3}$ per costruire un esempio di miglioramento dell'algoritmo. Questa modifica dovrebbe risolvere il problema degli arancioni e dei gialli e trasforma le linee di confusione in questo modo:

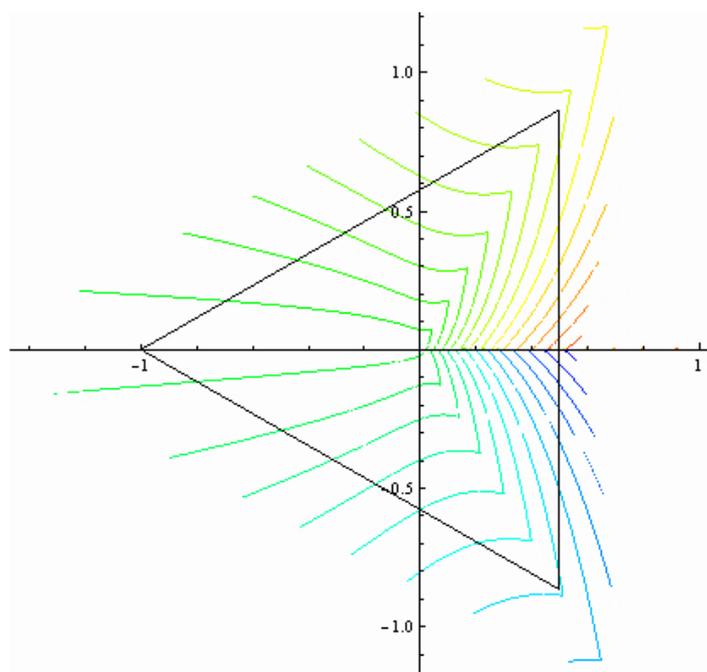


Figura 5.6: Possibile miglioramento.

Notiamo che in tutto il bordo del triangolo ogni punto è attraversato da una sola linea ed inoltre le linee sono abbastanza fitte. È evidente la differenza se confrontiamo con la figura a pagina 22: con il programma precedente le linee intersecavano il perimetro del triangolo in alto a destra con angoli quasi nulli, mentre in questo tutto il perimetro dovrebbe essere ben distinto.

5.1.6 Un errore trovato

Durante l'implementazione è stato riscontrato un errore. Esso sarebbe rimasto nascosto se non avessimo provato l'algoritmo su un'immagine, perché dal punto di vista teorico la funzione errata è coerente con ciò che volevamo. Soddisfaceva, cioè, la ridistribuzione dei colori rispetto alle linee di confusione e l'assegnazione della luminosità iniziale.

La funzione 5.2 della pagina precedente era scritta in questo modo, ottenuta ponendo $\alpha_4 + \beta_4 + \gamma_4 = I$:

$$\begin{cases} \alpha_4 = \frac{1}{3}(I + x_4 + \sqrt{3}y_4) \\ \beta_4 = \frac{1}{3}(I + x_4 - \sqrt{3}y_4) \\ \gamma_4 = \frac{1}{3}(I - 2x_4) \end{cases} . \quad (5.4)$$

In questo modo ogni punto del triangolo veniva portato nel cubo assumendo immediatamente la luminosità voluta, non era richiesta una seconda funzione (la 5.3). L'immagine trasformata dall'algoritmo era completamente insoddisfacente: aveva perso i colori vivaci ed era formata da colori grigiastri. Ora ricordiamoci che $x_4, y_4 \in [-1, 1]$, mentre $I \in [0, 765]$, e guardiamo attentamente la 5.4. Anche senza calcolare direttamente le maggiorazioni, si vede che α_4, β_4 e γ_4 sono uguali a I più una piccola quantità che dipende da x e y . Sono quindi quasi uguali tra loro e il colore $(\alpha_4, \beta_4, \gamma_4)$ risulta grigiastro⁸.

5.2 Risultati

Alcuni esempi di immagini ottenute sono nell'appendice da pagina 32. Si può notare immediatamente che il giallo resta giallo, mentre il verde diventa violaceo. Alcune tonalità di giallo diventano blu, come nel cartellino sul vaso di rose.

5.3 La sperimentazione

Per concludere il lavoro abbiamo sperimentato i risultati raggiunti. Le foto a pagina 32 e molte altre caratterizzate da simili ambiguità sono state mostrate ad un soggetto daltonico⁹. Egli ha visualizzato sul monitor prima le immagini naturali, poi quelle rimappate. Un suo primo commento è stato: 'Vedo cose che i daltonici non immaginano!'. Riferendosi alla pianta di rose, ci ha spiegato che nell'originale vede rose e foglie dello stesso colore, pavimento e parete dello stesso colore, rose e parete quasi dello stesso colore. In questa situazione riuscire a vedere i fiori risultava impossibile, come se ci fosse chiesto di trovare un bicchiere di acqua liscia fra mille di acqua frizzante. Possiamo cercare dove non ci sono bollicine, ma è molto difficile. Lui infatti cercava qualcosa del colore delle foglie, ma di forma un poco diversa. Nella foto accanto tutto risulta evidente e distinguibile: fiori, foglie, parete, pavimento. È come se nel bicchiere da cercare ci fosse vino anziché acqua. Anche nell'immagine dei Lego lui non riusciva a distinguere, per esempio, il prato dagli

⁸Per definizione di grigiastro. Vedere pagina 8

⁹Si tratta di un dicromatico, con cecità al verde.

omini che ci camminano. Ma anche alcune tonalità di grigio erano indistinguibili da altri oggetti. Anche qui, come in moltissime altre immagini, sia naturalistiche che artificiali¹⁰, la distinzione dei colori aumentava in modo soddisfacente. In una foto di papaveri nel campo di grano, invece, la rimappatura rendeva il campo a chiazze gialle e blu, a seconda di quanto era maturo (o verdognolo) il grano. Questo è dovuto alla discontinuità della funzione di rimappatura che si trova nella zona del giallo.

Dobbiamo comunque fare un'osservazione importante. Sappiamo benissimo dal punto di vista sperimentale cosa è migliorato, perché ci viene riferito dal soggetto. Però egli non può in nessun modo dirci cosa invece continua a confondere, perché non lo sa. Il mondo gli appare normale, con pochi colori. L'algoritmo gli permette con soddisfazione di distinguere molti più colori, ma ce ne sono altri che sono uguali ai suoi occhi.

Altri commenti sono stati fatti. Egli considerava molto artificiale la colorazione dell'immagine finale. Trovava piacevole guardare prima l'immagine trasformata, poi cercare nella foto originale quei particolari svelati. Finalmente riusciva a vedere le ciliegie sull'albero e le fragole sulle piante!

¹⁰Ad esempio le immagini a chiazze del test Ishihara, usate per riconoscere e una persona daltonica.

Capitolo 6

Conclusione

In questa tesi ho implementato un algoritmo di rimappatura dei colori tendente a disambiguare le coppie di colori vivaci normalmente confuse da un daltonico.

Ho anche condotto una prima sperimentazione con esito positivo. Il risultato è un algoritmo abbastanza efficace, ma che vuole essere solo un punto di partenza. Per un futuro necessario sviluppo ci si dovrà concentrare sulla taratura per adattare l'algoritmo ai diversi tipi di anomalia e ai diversi tipi di monitor.

Appendice A

Figure a Colori

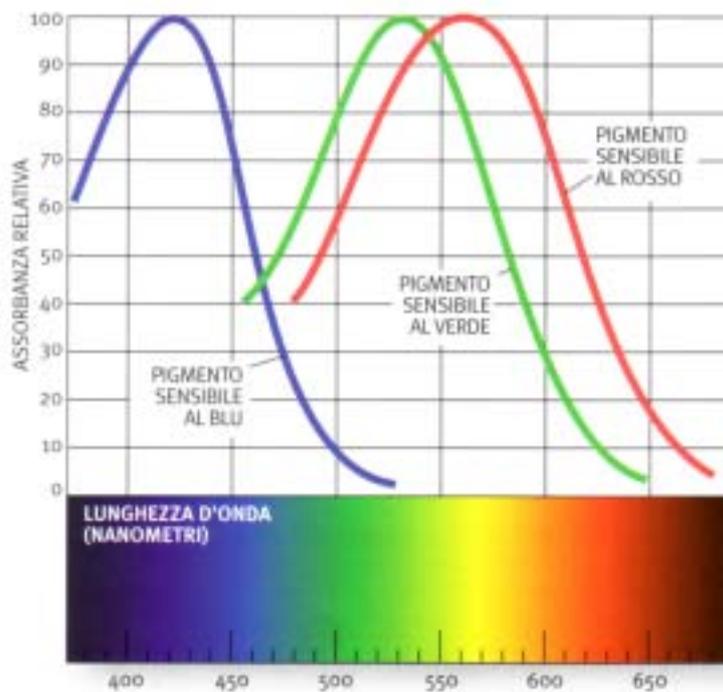


Figura A.1: La sensibilità dei coni alla lunghezza d'onda della luce.

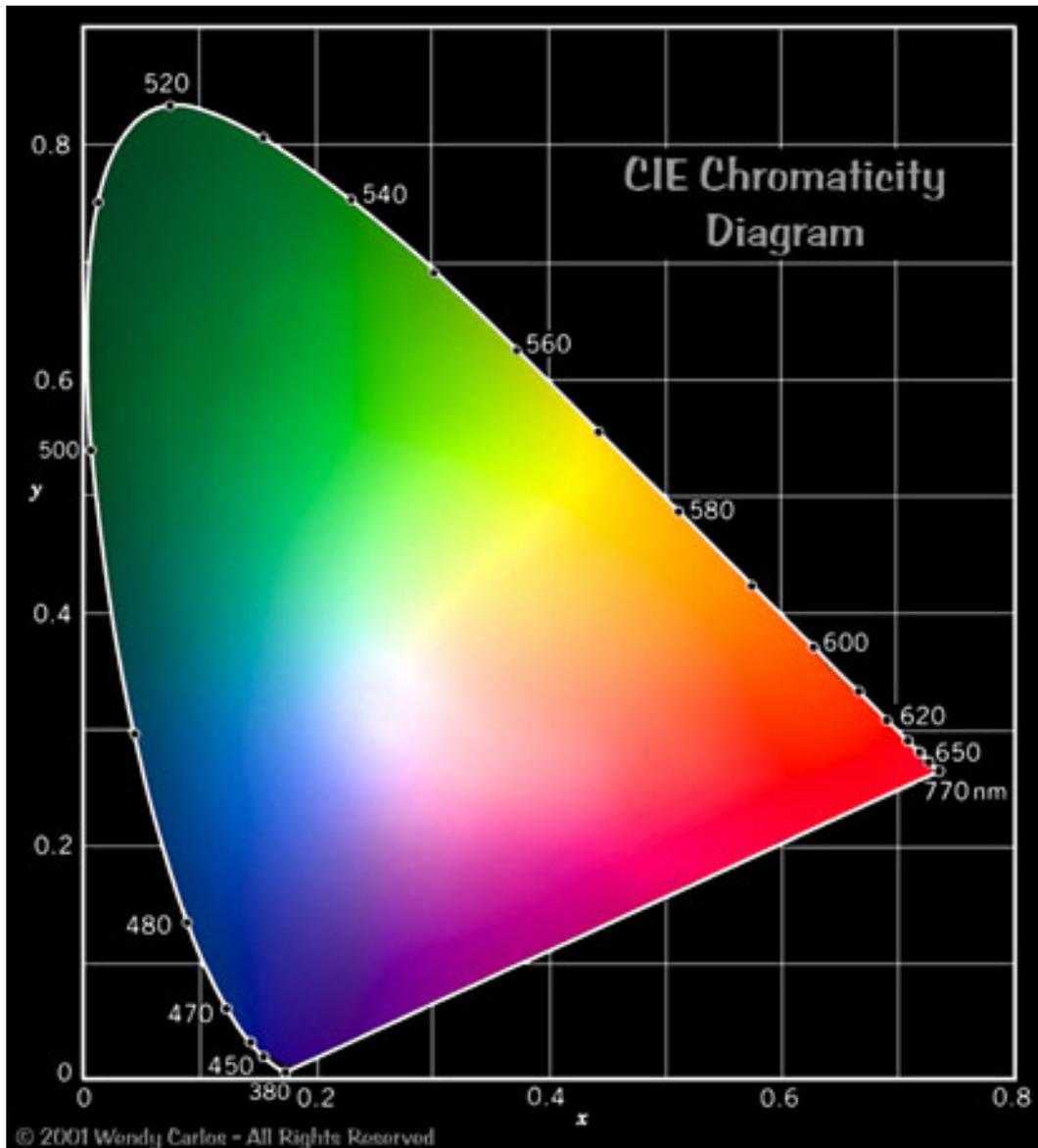


Figura A.2: Grafico cromatico CIE dei colori visibili dall'uomo.

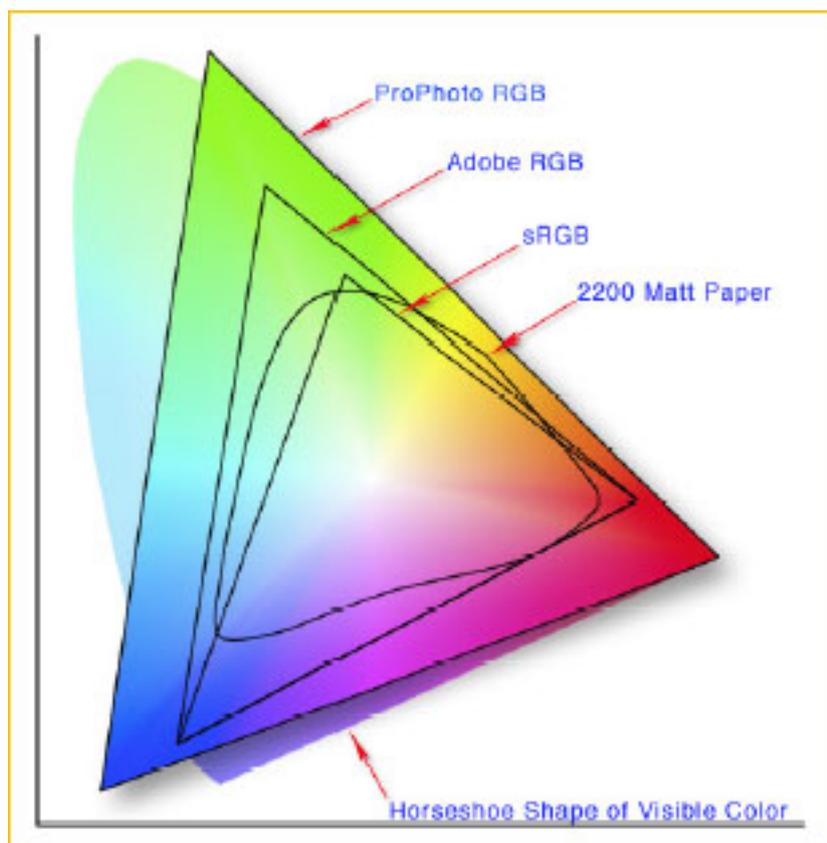


Figura A.3: Ci sono diversi tipi di schermo a cristalli liquidi. I colori riproducibili da ciascuno schermo sono all'interno dei triangoli. Lo sfondo con il bordo curvo contiene i colori visibili da noi (è il grafico CIE).

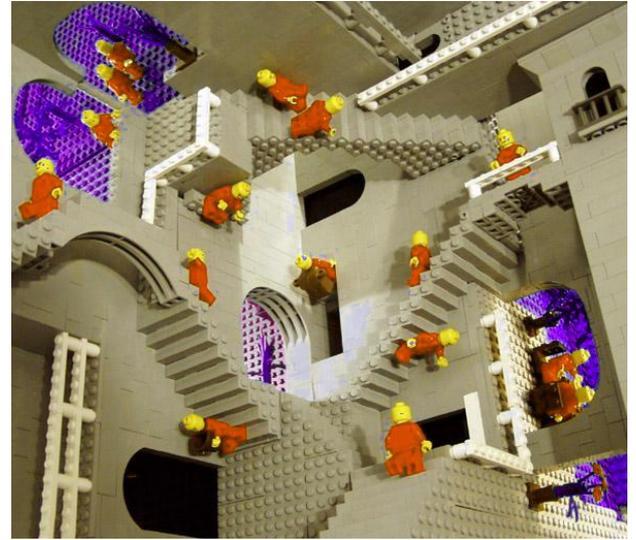


Figura A.4: L'immagine iniziale è a sinistra e la sua rimappatura a destra.

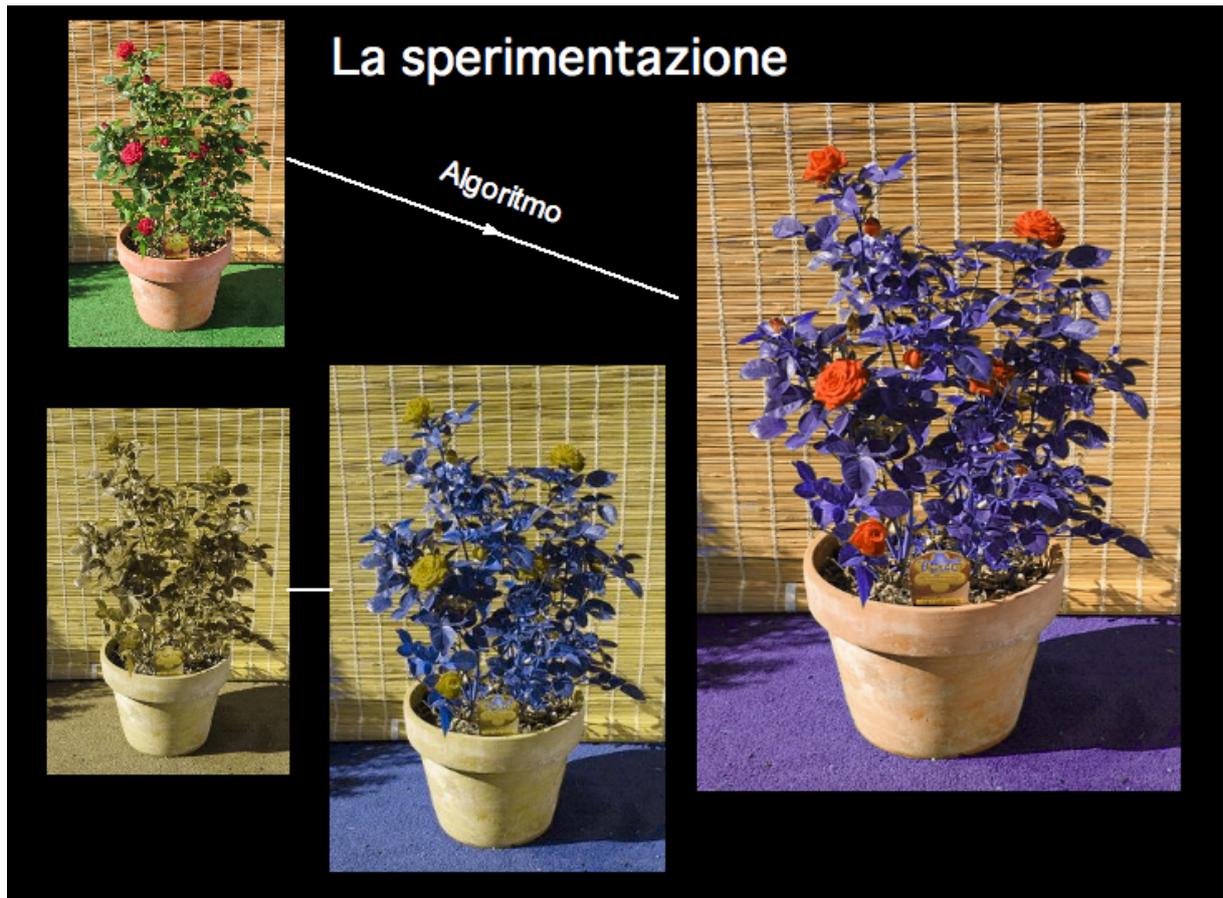


Figura A.5: In alto a sinistra: foto originale; alto a destra immagine rimappata; sotto: le rispettive simulazioni di visione di un daltonico, con Vischeck. Per l'immagine in basso a destra si è prima applicato l'algoritmo, poi la simulazione.



Figura A.6: Altri esempi di immagini rimappate.

Appendice B

Sorgente per Mathematica

Quest'appendice contiene il testo necessario per disegnare i grafici di pagina 20.

```
at = AbsoluteTime[];
x0 = 0.60;
y0 = -1;
numlinee = 60;
a = 0.0001 * numlinee; (*per eliminare le linee di troppo rendendole bianche*)
b = 1 - 0.001numlinee;

x:=(Mod[t, 1] - y0) Cos[Quotient[t, 1]π/2/numlinee + π/2] + x0 (*segmenti iniziali*)
y:=(Mod[t, 1] - y0) Sin[Quotient[t, 1]π/2/numlinee + π/2] + y0

z:=x + iy;

θ0:=Arg[z]
θ:=If[0 ≤ θ0 ≤ π, θ0, If[θ0 < 0, 0, π]];

r1:=If [0 ≤ θ < π/3, 2x, Abs [√3y - x]] (*triangolo → semocerchio*)

x1:=r1Cos[θ];
y1:=r1Sin[θ];

θ1:=If[0 < θ < π, 2θ, If[θ < 0, 0, 2π]]; (* semicerchio → cerchio*)
x2:=r1Cos[θ1];
y2:=r1Sin[θ1];
```

```

r2:=If [0 < θ1 < π/3 || 5π/3 < θ1 < 2π,  $\frac{r1^2}{2x^2}$ , If [π/3 < θ1 < π,  $\frac{r1^2}{\text{Abs}[\sqrt{3}y^2-x^2}]}$ ,  $\frac{r1^2}{\text{Abs}[\sqrt{3}y^2+x^2}]}$ ]];
(*cerchio→ triangolo*)
x3:=r2 Cos[θ1];
y3:=r2 Sin[θ1];

Show[
ParametricPlot[{x, y}, {t, 0, numlinee - 10^ - 7},
PlotPoints → (41numlinee),
ColorFunction → Function[{x, y, p}, If[Mod[pnumlinee, 1] < a || b < Mod[pnumlinee, 1],
White, Hue[p]]],
, RegionFunction → Function [{u, v}, u + √3v + 1 > 0 && u < 0.5 && u - √3v + 1 > 0 && 0 ≤ v]],
ListLinePlot [{{0.5, √3/2}, {-1, 0}, {0.5, -√3/2}, {0.5, √3/2}}, PlotStyle → Black]
]

Show[
ParametricPlot[{x1, y1}, {t, 0, numlinee - 10^ - 5},
PlotPoints → (41numlinee),
ColorFunction → Function[{x, y, p}, If[Mod[pnumlinee, 1] < a || b < Mod[pnumlinee, 1],
White, Hue[p]]]],
Graphics[Circle[{0, 0}, 1]]
]

Show[
ParametricPlot[{x2, y2}, {t, 0, numlinee - 10^ - 5},
PlotPoints → (41numlinee),
ColorFunction → Function[{x, y, p}, If[Mod[pnumlinee, 1] < a || b < Mod[pnumlinee, 1],
White, Hue[p]]]],
Graphics[Circle[{0, 0}, 1]]
]

Show[
ParametricPlot[{x3, y3}, {t, 0, numlinee - 10^ - 5},
PlotPoints → (41numlinee),
ColorFunction → Function[{x, y, p}, If[Mod[pnumlinee, 1] < a || b < Mod[pnumlinee, 1],
White, Hue[p]]],
(*, RegionFunction → Function [{u, v}, u + √3v + 1 > 0 && u < 0.5 && u - √3v + 1 > 0] *)],
ListLinePlot [{{0.5, √3/2}, {-1, 0}, {0.5, -√3/2}, {0.5, √3/2}}, PlotStyle → Black]
]
tempo impiegato (secondi):
N[AbsoluteTime[] - at, 3]

```

Bibliografia

- [1] Donald McIntyre *Colour Blindness: causes and effects*, Dalton Publishing;
- [2] Goldsmith T.H. *Come vedono gli uccelli* , Le Scienze n 460 - dic. 2006
Spiega l'evoluzione nella visione dei colori;
- [3] Oliverio Alberto, *La Mente*, BUR 2004 (capitolo 2);
- [4] Sacks Oliver, *L'isola dei senza colore*, Adelphi 1997;
- [5] Elettronica e telecomunicazioni, n°2 - agosto 2002 (p 32-37)
Display a cristalli liquidi: LCD, DSTN, TFT;
- [6] www.colourvision.info - Occhiali per daltonici;
- [7] gmazzocato.altervista.org/colorwheel/wheel.php
Aiuta a costruire pagine web leggibili da tutti;
- [8] colorhelper.com/tutorial.php - sito di EyepilotTM;
- [9] www.vischeck.com/vischeck
Trasforma le immagini simulando la visione di un daltonico cieco al verde.