4672: ESTUDI PER A UNA INTERFÍCIE D'AJUDA A DALTÒNICS EN LA INTERPRETACIÓ DE MAPES DE METRO.

Memòria del Projecte Fi de Carrera
d'Enginyeria Informàtica
realitzat per

Verónica Marchán Sánchez

i dirigit per

Maria Vanrell Martorell

Bellaterra, Juny de 2013

Universitat Autònoma de Barcelona

etse
Escola Tècnica Superior d'Enginyeria

La sotasignant, *Dra. Maria Vanrell Martorell*
Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per la *Verónica Marchán Sánchez*.

I per tal que consti firma la present.

Signat:.....

Bellaterra, Juny del 2013

Resum

En aquest projecte fem un estudi de diferents mètodes per a la segmentació i extracció de línies de mapes de metro com a suport per a daltònics. Hem aplicat dos mètodes amb intervenció de l'usuari i cinc mètodes automàtics on fem servir K-means per a la segmentació de color i Hough per a l'extracció de línies. Dels mètodes amb intervenció obtenim millors resultats amb un mètode d'assignació aproximada del color, i entre els automàtics tenim com a millor una solució ad-hoc sense paràmetres aplicada sobre l'espai RGB. D'acord amb els resultats experimentals, aquests mètodes ens permeten fer una bona segmentació i extracció de les línies de metro.

Resumen

En este proyecto hacemos un estudio de diferentes métodos para la segmentación y extracción de líneas de mapas de metro como soporte para daltónicos. Hemos aplicado dos métodos con intervención y cinco métodos automáticos donde hacemos servir K-means para la segmentación de color y Hough para la extracción de líneas. De los métodos con intervención obtenemos mejores resultados con un método de asignación del color, y entre los automáticos tenemos como mejor una solución ad-hoc sin parámetros aplicada al espacio RGB. De acuerdo con los resultados experimentales, estos métodos nos permiten hacer una buena segmentación y extracción de las líneas de metro.

Abstract

In this project we study different methods for segmentation and extraction of underground line maps as assistance for color-blind people. We have applied two methods with intervention and five automatic ones in which we use K-means for color segmentation and Hough for lines extraction. From the methods with intervention we get the best results with a color mapping method, and from the ones without intervention we have an ad-hoc solution with no parameters applied to the RGB color space as the best solution. According to the experimental results, these methods allow us to make a good segmentation and extraction of underground lines.

Índex

Capítol 1

Introducció

El daltonisme és un defecte genètic que comporta la dificultat per veure o distingir els colors. Aquest no és un defecte que condicioni de manera important la vida quotidiana d'un daltònic, però en algunes ocasions o en algunes professions pot comportar problemes com, per exemple, un pintor escollir els colors per a pintar un quadre. En aquest projecte hem desenvolupat un estudi d'una interfície d'ajuda en la visualització de mapes de metro que pugui servir de guia a l'hora de fer un seguiment d'una de les línies, les quals a vegades, i sobretot quan hi ha molts colors semblants, els pot esdevenir una tasca complexa.

En aquest capítol introduïm el problema del daltonisme i els problemes que pot originar en la comprensió de mapes de metro. A continuació, fem una breu descripció d'altres projectes que tracten de facilitar la discriminació dels colors per a que siguin comprensibles per a un daltònic i, finalment, expliquem els objectius que es volen assolir en aquest projecte.

1.1 El daltonisme

L'ull és l'òrgan encarregat de la percepció del nostre entorn mitjançant estímuls visuals. La seva arquitectura és realment complexa i especialitzada producte de milions d'anys d'evolució.

El globus ocular posseeix tres embolcalls: túnica fibrosa externa o escleròtica, túnica vascular mitjana o coroides i retina o túnica neural.

La túnica fibrosa externa és gruixuda i molt resistent, quasi inextensible. Es divideix en dues parts: la posterior, la escleròtica, i la anterior, la còrnia. La escleròtica és la veritable membrana de protecció de l'ull. La còrnia és una membrana transparent, circular situada a l'obertura exterior de la escleròtica. Aquesta, juntament amb el cristal·lí i la càmera anterior, refracta la llum. La coroides és una membrana que conté nombrosos vasos sanguinis i teixits cognitius. La funció d'aquesta capa és la de mantenir la temperatura constant i nodrir algunes estructures del globus ocular. La

retina es la capa més interna i complexa composta sobretot de cèl·lules nervioses. Les cèl·lules receptores sensibles a la llum o fotoreceptors es troben a la part exterior de la retina. Aquests fotoreceptors s'anomenen cons i bastons i son sensibles a diferents tipus de llum. Els bastons detecten les diferents intensitats i presenten una gran sensibilitat a la llum. Els cons son els responsables de la visió dels colors. Existeixen tres tipus de cons: els sensibles a la llum vermella, els sensibles a la llum blava i els sensibles a la llum verda.

El grau d'afectació de daltonisme pot oscil·lar entre una ceguera total del color o una simple dificultat en discernir alguns matisos entre colors.

Hi ha diferents tipus de daltonisme:

- *Acromàtic.* Aquest es un daltonisme molt poc habitual i és aquell on només es perceben els colors blanc i negre. L'individu no disposa de cap dels tres tipus de cons o no son funcionals.
- *Monocromàtic.* L'individu només té un dels tres tipus de cons, el que fa que ho vegi tot d'un mateix color, com per exemple el verd.
- *Dicromàtic.* En aquest cas, es sofreix una disfunció d'un dels tres mecanismes bàsics del color. Hi ha tres tipus:
  - *Protanopia.* Consisteix en la absència dels fotoreceptors del color vermell.
  - *Deuteranopia.* En aquest cas, es tracta de la falta de fotoreceptors del color verd.
  - *Tritanopia.* Aquesta és molt poc freqüent i consisteix en la absència dels fotoreceptors del color blau.
- *Tricromàtic anòmal.* L'individu, en aquest cas, té els tres tipus de cons, però defectuosos, cosa que fa que percebin els tons dels colors alterats. La majoria dels daltònics estan concentrats en aquest últim grup. En aquest es poden diferenciar tres subgrups:
  - *Protanomalia.* Aquells que confonen els vermells amb altres colors.
  - *Deuteranomalia.* Els que confonen els tons de verd amb altres colors.
  - *Tritanomalia.* Els que confonen els colors blaus amb altres colors.

Hi ha proves específiques per a detectar aquest tipus d'anomalia. El més conegut és el test de Ishihara, on a l'individu se li presenten un conjunt de cartolines amb cercles de diferents colors i que amaguen dins un número que han de saber distingir de la resta i que una persona amb una visió normal del color veuria.

El principal problema amb el que es troben els daltònics és en la dificultat de percebre aspectes de la realitat que per a individus normals no els és cap problema. En la majoria dels casos, ser daltònic no és un problema greu i no els afecta més

enllà de fer combinar la roba, o saber si la llum d'un aparell electrònic està verda o vermella, com els carregadors de les bateries, els ordinadors o els televisors. Problemes més greus tenen aquelles persones que en el seu àmbit laboral han de treballar amb el color, com son els dissenyadors, pintors, aviadors, astronautes o aquells que es dediquen a l'anàlisi del color, per exemple.

1.2 Mapes de metro i daltonisme

El color està carregat d'informació i constitueix una valiosíssima font de comunicadors visuals. Per això, si el propòsit de la imatge és comunicar alguna cosa que només pot fer a través dels colors, els daltònics no sempre podran comprendre tot allò que els hi volen transmetre. Aquest és el cas dels mapes de metro, on les rutes es distingeixen només pel color de les línies. Els mapes de metro varien segons les ciutats, però tots tenen la mateixa composició i el mateix objectiu. El problema que troben els daltònics és que, en molts casos, aquests mapes poden contenir línies amb colors que una persona amb visió normal del color pot discernir, però que per a un daltònic li pot resultar confós i, fins i tot, pot pensar que es tracta de la mateixa línia, trobant-se incapaços de distingir unes de les altres o simplement seguir el curs d'una d'elles. A la figura 1.1 podem veure un exemple.

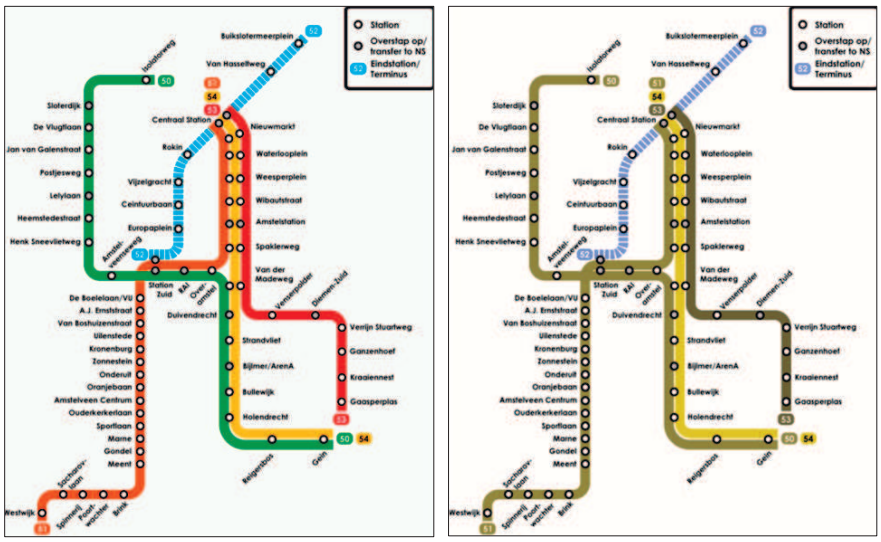


Figura 1.1: **Simulació de la imatge d'un mapa de metro per a un daltònic amb protanòpsia.** *(a) Imatge original del mapa de metro de Amsterdam. (b) Simulació, fent servir una eina de Photoshop, del mateix mapa tal i com ho veu un individu amb protanòpsia.*

Per aquest motiu, ens centrem en aquest tema i intentem solucionar el problema amb un estudi d'una interfície que ajudi als daltònics en aquesta tasca, independentment del tipus de daltonisme que sigui.

1.3 Antecedents

Actualment, hi ha tres aplicacions de propòsit general que són una ajuda pels daltònics a la visualització i la comprensió de les imatges:

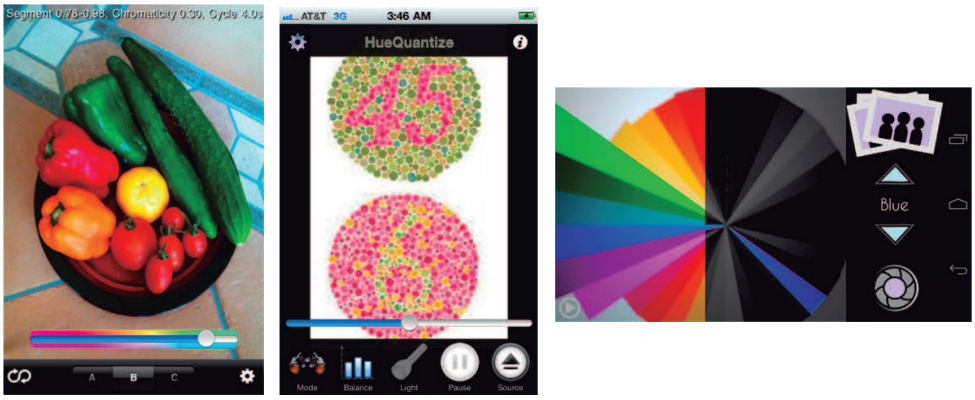
- *Cromatic Glass* [3]. Es tracta d´una aplicació per iPhone creada l'any 2010 per Kazunori Asada. Aquesta permet que els daltònics de tipus protanops, deuteranops i tritanops puguin identificar millor els colors dels objectes que veuen en temps real a través de la càmera del telèfon mòbil. Aquests colors es divideixen en diferents segments d'espectre de color segons el grau d'afectació cromàtica de l'usuari de manera que evita la superposició de colors.

- *DanKam* [9]. Dan Kaminsky va crear aquesta aplicació l'any 2010, per a iPhone i Android. Es tracta d'una aplicació de realitat augmentada. Ajuda als usuaris a buscar els colors, jugant amb ells i la saturació en temps real, que els sigui més fàcil de detectar, diferenciar i permetre comprendre millor el que veuen. Aquesta es una aplicació centrada en el tipus de daltonisme més comú, tricromàcia anòmala. El primer que hem pogut observar fent servir l'aplicació és que és l'usuari qui ha de buscar la combinació de colors que li permeti entendre el que està veient. En el cas dels mapes de metro, l'usuari es pot trobar que cap de les combinacions li ajudi, doncs quan ha aconseguit veure bé una línia, potser algunes de les altres han canviat de color i ara les veu del mateix, de manera que no les pot distingir.

- *Colorblind Vision* [7]. Aquesta és una aplicació creada per Bradley C. Grimm. Dins d'aquesta aplicació ens trobem amb diverses opcions, una d'elles és l'extracció de colors. Aquesta secció el que fa és extreure alguns dels colors bàsics, concretament el groc, taronja, vermell, magenta, púrpura, blau, cian i verd, de la imatge. El problema que trobem en aquesta aplicació és que, en el cas de tenir un mapa de metro amb línies de colors que no processa l'aplicació, ja no les detecta.

Podem veure el funcionament d'aquestes aplicacions a la figura 1.2.

Algunes d'aquestes aplicacions no són automàtiques, de manera que l'usuari ha d'anar buscant la combinació de colors que li vagi bé per poder comprendre la imatge. Tampoc són aplicacions específiques a un problema concret, de manera que en molts casos no funcionen correctament. Per aquests motius, aquestes aplicacions no ens son útils en la segmentació de les línies de mapes de metro.



(a) (b) (c)

Figura 1.2: **Aplicacions per daltònics.** *Es donen diverses captures de pantalla de les aplicacions existents. (a) Cromatic Glass (canvi de la cromaticitat del verd), (b) DanKam (modifica el matís de la imatge), (c) ColorBlind Vision (extracció del color blau).*

1.4 Objectius del projecte

L'objectiu principal d'aquest projecte és desenvolupar algoritmes que processin la imatge de mapes de metro i facilitin la seva visualització a usuaris amb qualsevol tipus de daltonisme. Ens centrarem en l'estudi d'un sistema totalment automàtic per aconseguir fer una segmentació i extracció correcta de les línies de metro sense cap intervenció, tot i que també farem un estudi previ d'intervenció manual amb l'usuari.

En tots dos casos, com veurem en capítols posteriors, el problema que hem de resoldre per a portar a terme totes aquestes interaccions és el de la segmentació i classificació de les diferents línies del mapa de metro de qualsevol ciutat.

La primera dificultat amb la que ens trobem a l'hora de fer la segmentació és el de diferenciar les línies de metro del text i del fons de la imatge. El segon problema és poder donar nom a aquests colors y el tercer problema, i el principal, és aconseguir fer una correcta diferenciació dels colors de les línies. Els mapes de metro poden contenir línies de diferents colors i mida, els quals poden variar entre uns mapes i altres i, fins i tot, dins del propi mapa de metro. Les línies poden tenir colors molt semblants, sobretot en els mapes de metro amb moltes línies. A més, les imatges no sempre son de bona qualitat. Ens trobem, en la majoria dels casos, amb imatges amb anti-aliasing.

Finalment, en la figura 1.3 tenim englobat l'objectiu final en un esquema. En aquest projecte només ens hem dedicat a fer la part de segmentació del color i de l'extracció de les línies de metro. Hem fet servir imatges digitals, no extretes de la càmera d'un dispositiu mòbil sinó d'Internet, que tenen el color ben balancejat. El problema que hem de resoldre és suficientment complex per a fer aquesta simplificació

i extreure resultats significatius que ens permetin fer una avaluació dels mètodes emprats. Per aquest motiu, el pre-processament de la imatge es deixa com a tasca futura en el que s'haurien d'aplicar algoritmes de constància de color.

En el mòdul d'interpretació del mapa tindrem en compte dos casos: Amb intervenció de l'usuari, on l'usuari fa una selecció de la línia que vol resseguir i aquesta li surt marcada, i sense intervenció, on automàticament es fa la segmentació i l'extracció de les línies.

En un principi, la idea del projecte era arribar a fer una aplicació per a un dispositiu mòbil on dur a terme el mòdul de visualització adaptada a l'usuari. Per falta de temps no ha estat possible arribar a completar aquesta part, però s'ha fet una aproximació de la visualització al capítol 4.

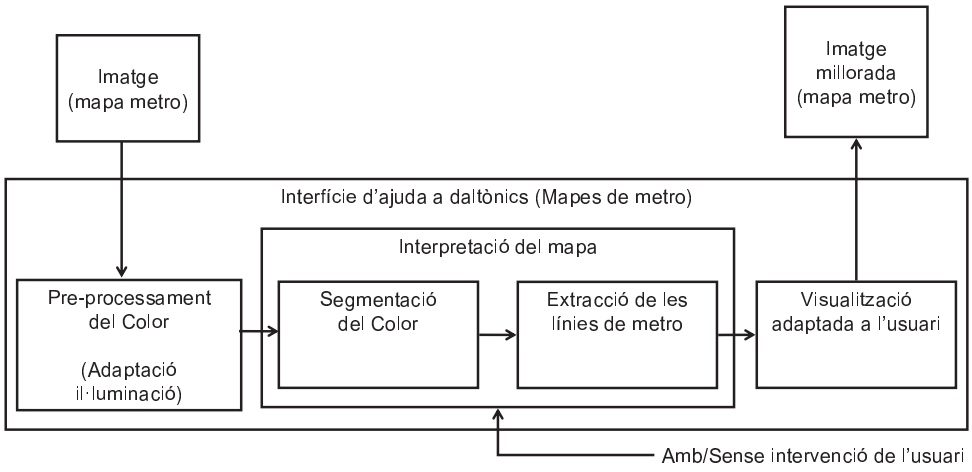


Figura 1.3: **Esquema del projecte.**

Capítol 2

Interpretació automàtica de mapes

L'anàlisi d'imatges compren els mètodes i tècniques que es fan servir per a extreure informació relativa a un objecte d'estudi a partir d'una imatge. Una de les tècniques més utilitzades en l'anàlisi d'imatges és la segmentació, que es fa servir tant per localitzar objectes com per a trobar els límits d'aquests dins de la imatge. La segmentació d'una imatge consisteix en dividir la imatge en regions homogènies i disjuntes a partir d'un conjunt de característiques, com el color, la intensitat, la textura, la magnitud del gradient o la direcció dels contorns, que permeten discriminar unes zones d'altres i així poder analitzar-les més fàcilment.

En aquest projecte, l'objecte que volem segmentar i extreure son les línies dels mapes de metro. Les característiques que fem servir per fer la separació de les regions de la imatge és el color de les línies i, en alguns casos, la detecció mateixa d'aquestes.

En aquest capítol expliquem els diferents mètodes que hem fet servir per a fer la segmentació i extracció de les línies de metro. En la primera part, fem una descripció dels mètodes de segmentació amb intervenció de l'usuari. Aquesta intervenció serà la selecció de la línia que l'usuari vol segmentar. En la segona part, expliquem els algoritmes que s'han fet servir per a la segmentació i extracció de les línies dins de dos descriptors de color diferents: Color Naming i RGB. A continuació, expliquem quin dels dos descriptors utilitzem al llarg dels mètodes. Finalment, fem un seguiment dels mètodes que fem servir per intentar resoldre el problema que trobem amb el paràmetre de la segmentació.

2.1 Amb intervenció de l'usuari

A les aplicacions que hem comentat en el capítol anterior, la intervenció de l'usuari es fa necessària per a que pugui ajustar la imatge a les seves necessitats. Molts cops,

aquesta intervenció pot suposar una pèrdua de temps per part de l'usuari a l'intentar buscar els colors adequats per a ell, i no sempre és una tasca fàcil. Com ja em vist, quan tenim un mapa amb moltes línies, resulta molt difícil trobar una combinació de colors que facilitin la distinció dels mateixos. Tot i que la finalitat del nostre projecte és arribar a un mètode de segmentació automàtic, per a l'aplicació final hem volgut afegir una part d'intervenció amb l'usuari, però no amb la finalitat de que sigui ell qui ajusti els colors a la seva necessitat, sinó donant-li l'oportunitat de demanar, directament, allò que vol veure amb més claredat i ser l'aplicació qui li doni la solució. En aquest apartat l'única intervenció que hi ha és la de la selecció, per part de l'usuari, de la línia que vol visualitzar.

2.1.1 Mètode 1. Assignació directa de color

Dins del mètodes de segmentació amb intervenció de l'usuari, aquest és el més intuïtiu. La idea és fer una segmentació de la línia que volem que se'ns mostri mitjançant la selecció S d'un dels N píxels d'aquesta agafant el seu color (R_p, G_p, B_p). A continuació, buscar tots els píxels p_i, amb color $(R_{p_i}, G_{p_i}, B_{p_i})$, amb els mateixos valors de color segons l'equació

$$S(p) = \begin{cases} 1 & \text{si } R_p = R_{p_i}, G_p = G_{p_i}, B_p = B_{p_i} \\ 0 & \text{en qualsevol altre cas.} \end{cases} \tag{2.1}$$

Un cop trobats, ja podrem mostrar la imatge resultant tal i com veiem a la figura 2.1, on se'ns mostra un exemple de segmentació.

2.1.2 Mètode 2. Assignació aproximada de color

Tot i que el mètode anterior dona resultats bastant acceptables, ens trobem amb el problema de que, a les imatges amb línies molt primes, hi ha una degradació de colors a les voreres de les línies de manera que el valor del píxel seleccionat no coincideix amb els corresponents a la línia de metro que es vol segmentar. Per això, en el mètode 2 s'ha volgut anar més enllà i intentar arreglar aquests problemes.

El que s'ha fet és, en comptes de buscar el píxels de la imatge amb el mateix RGB que el píxel seleccionat, s'ha agafat aquest RGB i s'ha buscat en un rang de valors RGB molt propers a aquest. D'aquesta manera, aconseguim, tal i com podem veure a la figura 2.2, una bona segmentació d'aquelles línies que al mètode anterior no es segmentaven. Fins i tot, s'ha millorat la visualització d'aquelles línies que sí es segmentaven. L'equació que s'ha fet servir és

$$S(p) = \begin{cases} 1 & R_p - 50 \leq R_{p_i} \leq R_p + 50, G_p - 50 \leq G_{p_i} \leq G_p + 50, B_p - 50 \leq B_{p_i} \leq B_p + 50. \\ 0 & \text{en qualsevol altre cas.} \end{cases} \tag{2.2}$$

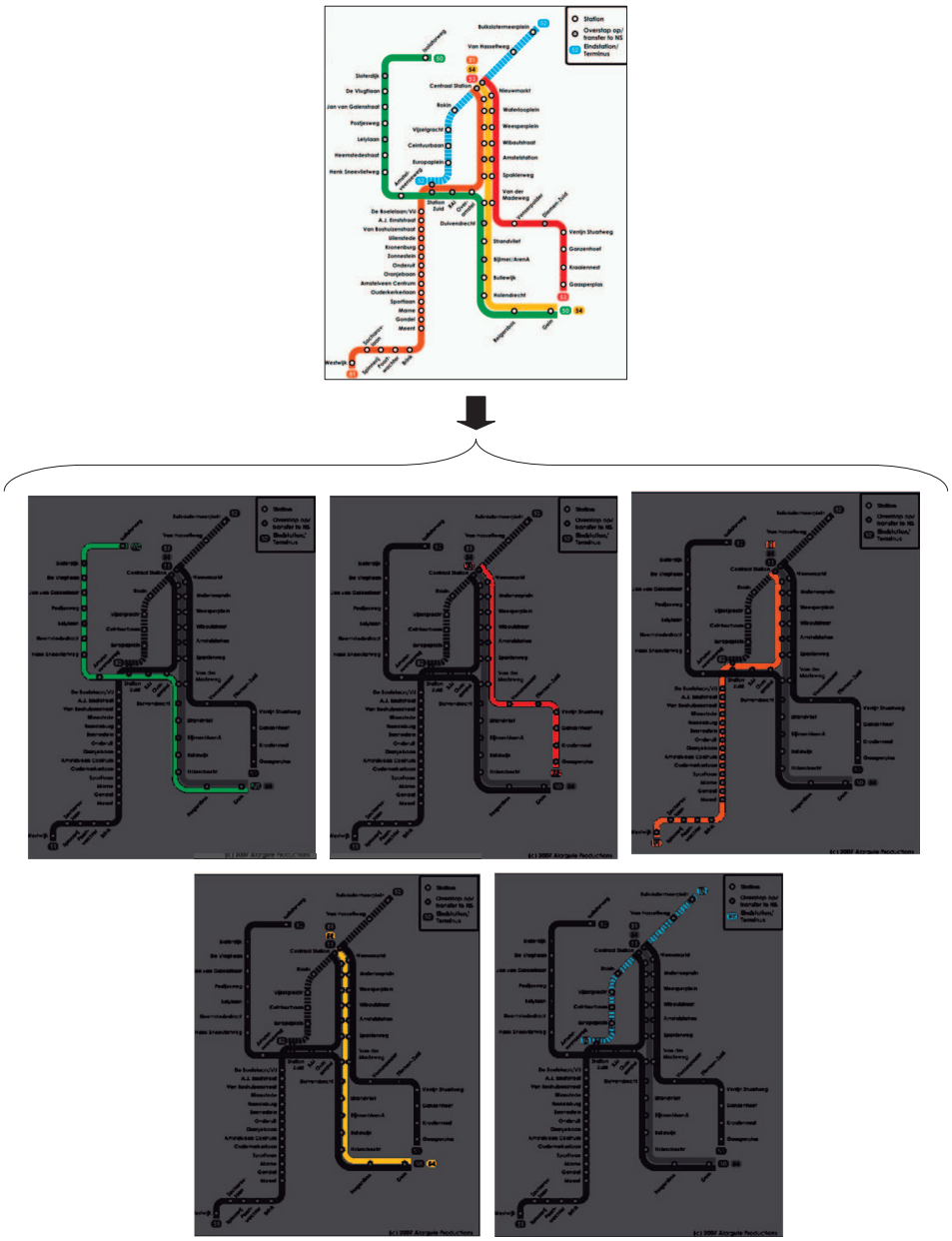


Figura 2.1: **Segmentació de les línies del mapa de metro de Amsterdam amb el mètode 1.** *Aquí tenim el resultat d'aplicar aquest mètode per a cadascuna de les línies. Partint de la imatge original de la part superior, seleccionem un dels seus píxels, fem una canvi a grisos de la imatge i pintem de color només els píxels que tinguin el mateix valor RGB que el píxel seleccionat.*

On R_p, G_p i B_p son els valors RGB del píxel seleccionat p, i R_{p_i}, G_{p_i}, B_{p_i} son els valors RGB de cada un dels píxels de la imatge i $i \epsilon 1..N$.

El rang de valors que hem fet servir en aquest mètode s'ha escollit experimentalment.

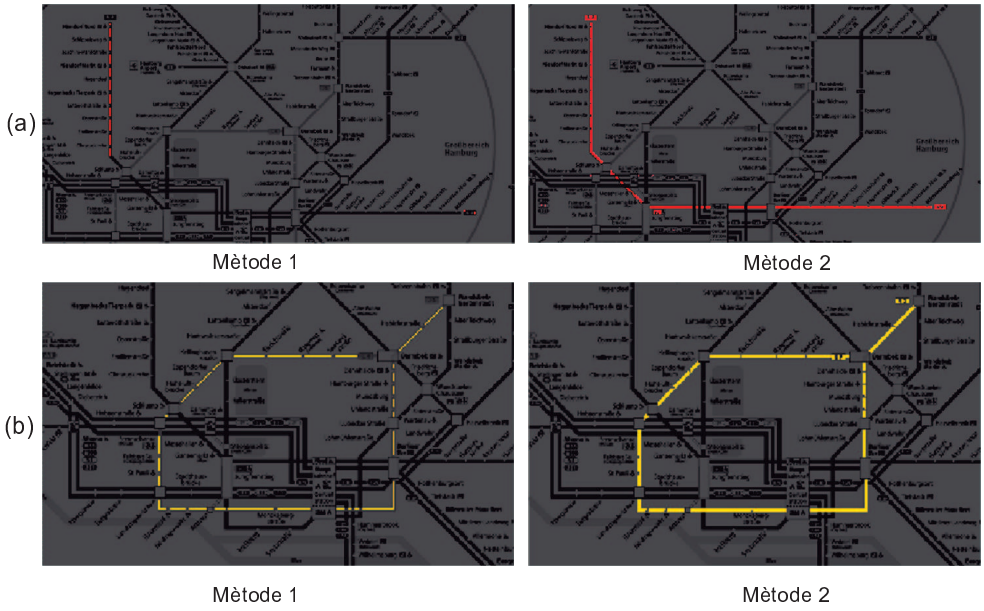


Figura 2.2: **Comparació entre el mètode 1 i el mètode 2.** *Les imatges de la fila (a) ens mostren com el mètode 2 ha segmentat correctament una línia que el mètode 1 no ha fet. Tal i com podem veure a les imatges de la fila (b), tots dos mètodes segmenten bé aquesta línia de metro, però en el cas del segon mètode es veu més clarament.*

2.2 Sense intervenció de l'usuari

Un dels objectius d'aquest projecte consisteix en aconseguir una correcta segmentació de les línies de metro sense necessitat d'intervenció. En aquesta secció expliquem uns mètodes que augmenten la complexitat progressivament tot buscant millorar els resultats i els algoritmes utilitzats en aquests mètodes per a la segmentació de color i l'extracció de les línies.

2.2.1 Segmentació de color: K-means

K-means [10] és un algoritme de clustering que ens permet fer la partició d'un conjunt d'N elements en K grups diferents entre ells. Cada un dels grups està definit per un

centroide, el qual ve donat per la mitjana dels seus membres. D'aquesta manera tenim l'avantatge de tenir cada grup caracteritzat pel seu centroide.

Inicialment escollim K centroides aleatòriament. A continuació, l'algoritme assigna cada element al centroide que té més a prop. Un cop assignats, calcula la mitjana de cada grup i aquesta serà la nova posició del centroide. Seguidament, torna a fer una assignació dels elements i es torna a fer un càlcul de la mitjana. Com que es tracta d'un algoritme iteratiu, aquests passos els fa fins que els centroides deixen de moure's, la qual cosa vol dir que l'algoritme a convergit.

Com es tracta d'un algoritme heurístic, no hi ha una garantía que convergeixin a l'òptim global i el resultat depèn de la posició dels centroides al inici. El que és pot fer és executar l'algoritme vàries vegades amb diferents condicions de partida. Això es pot fer perquè l'algoritme acostuma a ser ràpid, amb un temps d'execució polinomial, però en alguns casos pot arribar a tenir un temps d'execució exponencial per a arribar a convergir.

En el nostre cas, per garantir una estabilitat als nostres mètodes, hem executat l'algoritme 8 cops. Amb més iteracions el temps d'execució era massa gran.

L'algoritme K-means és el següent:

```
1.  Escollir K aleatòriament i inicialitzar les posicions dels
centroides.
2.  Assignar cada punt de la imatge al centroide més proper.
3.  Mentre no convergeixi:
4.    Per a cada clúster:
5.       Fer la mitjana dels seus elements.
6.       Moure el centroide a la seva nova posició.
7.    Tornar a assignar cada punt de la imatge al centroide més
proper.
8.  Retornar els clústers.
```

A l'hora d'aplicar K-means per a la segmentació de colors, tenim dos descriptors diferents amb els que podem treballar: Color Naming i RGB, els quals estudiarem a fons a la secció 2.2.3.

Uns dels aspectes que caracteritzen aquest algoritme és que el nombre de clústers, K, és un paràmetre d'entrada, i la mala selecció d'aquest valor pot fer que els resultats no siguin bons. En els mètodes automàtics no podem introduir manualment aquest valor, per tant hem de buscar la manera de que sigui automàtic i que s'adapti als diferents mapes de metro. Aquesta cerca la veurem a la secció 2.2.4.

2.2.2 Extracció de línies de metro: Transformada de Hough

La Transformada de Hough és un algoritme [6, 8] que es fa servir en reconeixement de patrons en imatges per trobar línies rectes, circumferències o el·lipses. En aquest

cas la fem servir per la detecció de línies en cadascuna de les imatges resultants del K-means de manera que puguem descartar totes aquelles que no contingui cap línia.

Per a detectar línies es fa servir l'equació de la recta en coordenades polars:

$$\rho = x \cdot cos\theta + y \cdot cos\theta \tag{2.3}$$

Amb això, es pot associar una recta a cada parella de punts (ρ, θ), el qual es denomina espai de Hough per el conjunt de rectes en dos dimensions.

Per a un punt de la imatge (x, y) qualsevol, les rectes que passen per aquest punt estan determinades per (ρ, θ), on ρ, que indica la distancia entre la línia i l'origen, queda determinat per θ segons l'equació 2.3. Això correspon a una corba sinusoïdal a l'espai (ρ, θ), que és única per a aquell punt. Si dues corbes corresponent s'intersequen, el punt d'intersecció a l'espai de Hough correspon a una línia. Per tant, l'únic que s'ha de fer és trobar corbes concurrents per a detectar punts col·lineals. A la figura 2.3 tenim un exemple.

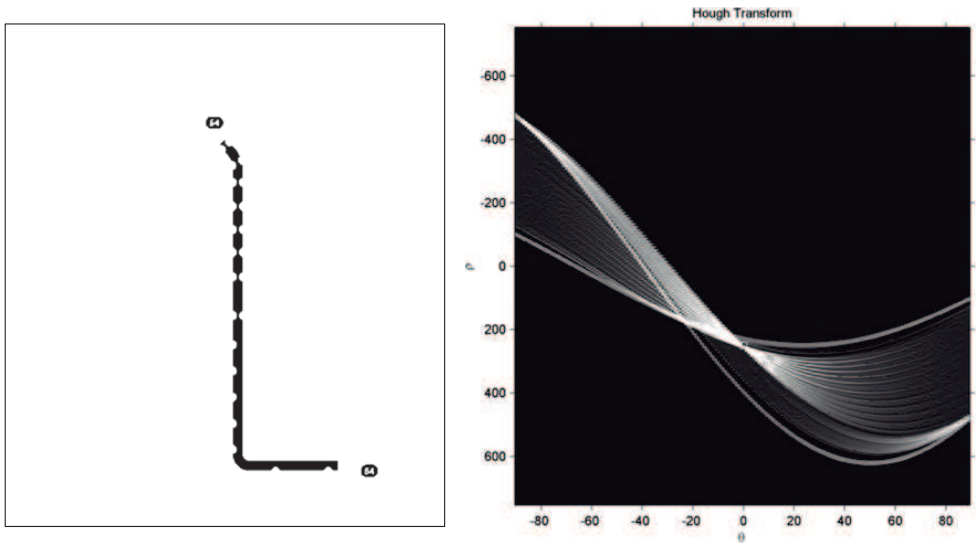


Figura 2.3: **Espai de Hough corresponent a una línia.** *Aquí tenim l'espai de Hough d'una línia determinada. Es tracta de corbes sinusoïdals, moltes d'elles són punts de la imatge que intersequen en un punt* (ρ, θ) *en l'espai de Hough, el qual correspon a una línia.*

La transformada de Hough retorna una matriu, la qual conté el nombre de vots per a cada conjunt de punts (x, y). Aquells que tinguin més vots respecte al nombre total de punts, contindran línies. Aquest algoritme és eficient si un gran número de vots cau a la posició correcte i la seva complexitat augmenta segons la quantitat de punts que hi ha a la imatge. La seva eficiència també depèn de la qualitat de les dades d'entrada de l'algoritme. Les voreres s'han de detectar bé per a que sigui eficient el

procediment. Fer servir la transformada de Hough en imatges amb molt de soroll acostuma a ser un problema i, generalment s'ha de tractar la imatge abans per a eliminar-lo.

A continuació, definim la transformada de Hough:

```
1.  Detectar contorns en la imatge.
2.  Per a cada punt de la imatge:
3.    Si el punt (x, y) està en un contorn:
4.       Per a tots els possibles angles θ:
5.          Calcular ρ per el punt (x, y) amb un angle θ
6.          Incrementar la posició (ρ, θ) a l'acumulador.
7.  Buscar les posicions amb els valors més grans a l'acumulador.
8.  Retornar les rectes que tenen valors més alts a l'acumulador.
```

2.2.3 Selecció de l'espai de color

Per a dur a terme la segmentació dels colors tenim dos descriptors o espais de color amb els que podem treballar i que hem estudiat per veure amb quin d'ells obtenim millors resultats.

La K que aplicarem a l'algoritme K-means en els següents dos mètodes s'ha calculat experimentalment de la mediana del nombre de colors que conté cada imatge de la base de dades (veure Capítol 3). En aquest cas, $K = 10$.

2.2.3.1 Mètode 3. L'espai dels noms de color

Els primers que van parlar del concepte de Color Naming van ser B. Berlin i P. Kay [5]. Estudiant un gran nombre de llengües existents al món, es van trobar que existeixen uns determinats noms de colors que són bastant universals i que es fan servir en la majoria de cultures. Les llengües més evolucionades (com l'anglès o l'espanyol) en tenen 11 i a mesura que trobem llengües menys evolucionades n'hi ha menys.

El concepte de Color Naming s'ha estudiat dins de la ciència del color durant molts anys i hi ha molta gent que ha intentat trobar quines eren les fronteres entre tots aquests noms en un espai de color de tres dimensions. El software amb la que hem treballat està basat en els resultats de la tesi de R. Benavente [4], en la que va desenvolupar el primer model que hi ha on, donat un color en RGB o CIELAB, et retorna un vector de probabilitats de que aquest color rebi unes determinades etiquetes de noms de color, que representem com un vector d'11 dimensions, com per exemple ($P_1 = blanc, P_2 = negre, ..., P_{11} = lila$). En aquest model es fan servir els següents colors bàsics: vermell, taronja, marró, groc, verd, blau, púrpura, rosa, negre, gris i blanc.

En aquest projecte partim de la hipòtesi que els mapes de metro s'han creat de tal manera que la gent s'hi pugui referir amb termes lingüístics (la línia vermella, la

línia lila, la línia rosa, etc.). Per això la primera hipòtesi del projecte ha estat intentar veure si podem segmentar una imatge de metro fent servir el descriptor de noms de color.

L'algoritme del Color Naming és el següent:

1. Càlcul nivells de lluminositat en el model.
2. Conversió de la imatge de RGB a CIELab
3. Assignació de tots els píxels al nivell de lluminositat corresponent.
5. Trobar valors de pertinença a categories cromàtiques: Calcular el valor de la funció TSE (Triple Sigmoid with Elliptical centre) per a cada valor del vector $s = [a, b]$ on a i b son valors calculats resultants de la imatge CIELab
6. Trobar valors de pertinença a categories acromàtiques: Calcular el valor de la funció sigmoïdal per a cada valor dels valors 1D de $x = L$.
7. Obtenim la imatge de sortida amb cada píxel marcat amb el color del màxim valor d'afiliació.
8. Obtenim el descriptor de color amb la membresia a totes les categories.

En aquest mètode, el primer que fem, un cop llegida la imatge RGB o CIELab, és aplicar la funció ColorNaming i treballar amb la imatge resultant fent servir l'algoritme K-means. Un cop tenim els clústers, mitjançant Hough, filtrem aquells clústers que no contenen línies i ens retorna aquells que considera que sí contenen.

Treballar amb la imatge que ens retorna la funció Color Naming (veure figura 2.4) ajuda a que els clústers estiguin més definits perquè la definició del color d'una línia és més pura, sense degradacions als contorns i tots els píxels de la línia tenen el mateix valor RGB. Aquest factor garanteix una bona separació dels colors de les línies.

Després d'aplicar aquest mètode, hem comprovat que la hipòtesi inicial, a la pràctica, no funciona, tal i com veurem als resultats dels experiments del capítol 3. Així doncs hem optat per treballar amb RGB.

2.2.3.2 Mètode 4. L'espai RGB

RGB es un model de color amb el que es pot representar un color mitjançant la barreja dels tres colors de llum primària.

El primer que fem en aquest mètode és llegir la imatge de la que volem extreure les línies de metro. Com que les imatges acostumen a tenir les línies sobre un fons blanc o un color pàl·lid, el que fem a continuació és treure el fons. En el nostre cas, tots els píxels amb $(R, G, B) > (128, 128, 128)$. Sobre aquesta imatge sense fons apliquem

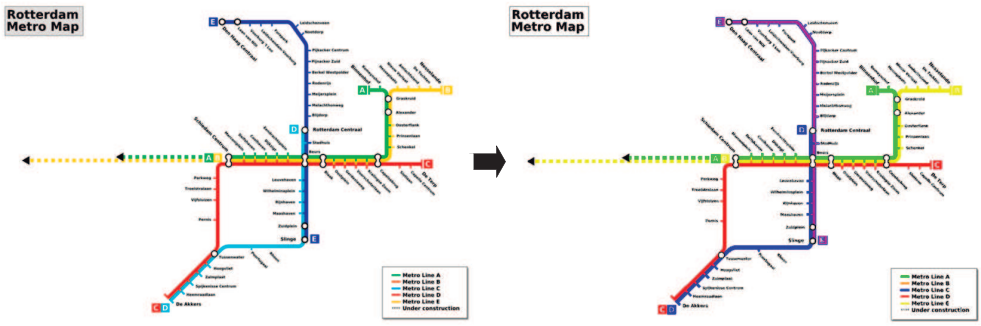


Figura 2.4: **Imatge resultant de la funció Color Naming.** *La imatge de l'esquerra és la imatge original i la imatge de la dreta és la que retorna la funció Color Naming. Com que dins dels colors bàsics no tenim distincions de tons de colors, els dos blaus de la imatge original els ha pintat de blau i lila.*

l'algoritme K-means.

Un cop aplicat aquest algoritme, per a cada clúster, creem una imatge negra amb els píxels del clúster en blanc. Com que un clúster pot contenir informació que no volem perquè no conté línies sinó soroll, a cada imatge que obtenim li apliquem la Transformada de Hough. Un cop aplicada, ja tenim els clústers corresponents a les línies del mapa de metro. Així doncs, construïm una imatge en color per a cadascun d'aquests clústers, com podem veure a la figura 2.5.

Com ja hem comentat al mètode 3, a l'hora d'aplicar la segmentació amb el descriptor RGB obtenim millors resultats. Per tant, el que farem servir en els següents mètodes serà aquest.

2.2.4 Selecció dels paràmetres

Com ja hem comentat anteriorment, el principal problema de la segmentació del color és trobar un valor adequat pel paràmetre K d'entrada de l'algoritme K-means. Una mala selecció d'aquest valor pot donar resultats molt dolents. La qüestió és que, com que volem que sigui totalment automàtic, aquest paràmetre s'ha de calcular prèviament segons les característiques de cada mapa de metro. En aquesta secció intentarem trobar un mètode que ens permeti donar amb la millor solució a aquest problema.

2.2.4.1 Mètode 5. Mesura de *Validity*

En els mètodes anteriors, a l'hora d'aplicar l'algoritme K-means ho fem indicant-li un nombre de clústers concret, el qual és el mateix per a cada imatge. Però aquest sistema és molt ineficient perquè inicialment no coneixem la quantitat de clústers que necessitem per a fer una bona segmentació de les línies d'un mapa de metro, i aquestes poden variar entre uns i altres. En el cas de tenir més línies que clústers, ens pot

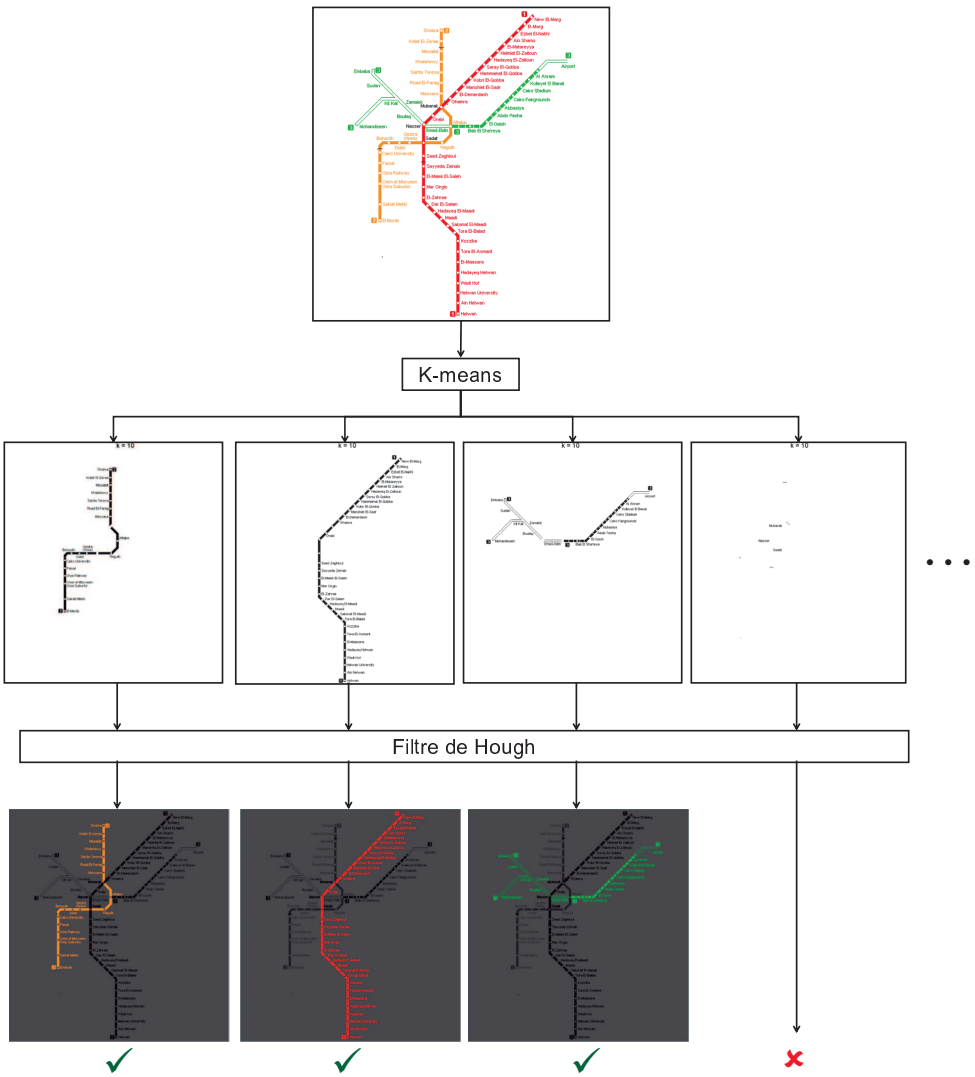


Figura 2.5: **Resultat del mètode 3.** *Quan apliquem K-means obtenim 10 clústers, els quals poden, o no, contenir els píxels corresponents a una línia. Podem veure que la transformada de Hough filtra alguns d'aquests clústers, de manera que les imatges resultats donin una correcta segmentació.*

agrupar vàries línies en un mateix clústers. Per aquest motiu, en aquest mètode hem fet unes variacions per a corregir aquest problema.

Hi ha moltes maneres de calcular el nombre de clústers reals. La que hem fet servir en aquest mètode ha estat la proposada per Siddheswar Ray i Rose H. Turi [11]. Es tracta de fer servir dues mesures. La primera, anomenada distància intra-clúster, fa servir les distancies dels punts al centre del clúster al que pertanyen per determinar com és de compacte. Aquesta mesura es defineix com

$$intra = \frac{1}{N} \sum_{i=1}^{K} \sum_{x \epsilon C_i} \parallel x - z_i \parallel^2$$

on N es el nombre de píxels de la imatge, K el nombre de clústers, x son les mostres i z_i es el centre del clústers C_i. Quant més petita sigui l'intra-cluster, més compacte serà el clústers.

La segona mesura que ens servirà per determinar la K és la distància inter-cluster. Aquesta és la distància mínima que hi ha entre els centres dels clústers, que definim com

$$inter = min(\parallel z_i - z_j \parallel^2)$$

on $i = 1, 2, ..., K-1$ i $j = i+1, ..., K$.

Un cop tenim aquestes dues mesures ja podem determinar si tenim un bon clustering combinant-les i calculant el ràtio de la següent manera:

$$validity = \frac{intra}{inter}$$

Com que no sabem *a priori* quantes línies tenim en un mapa de metro, hem triat un rang de valors per a fer aquests càlculs, $2 < K < 26$. El valor més petit de K és 3 perquè hem pensat que seria, a partir d'aquest número de línies, quan seria necessària una distinció entre elles. El valor més gran és 25 perquè no hi ha mapes de metro amb més línies que aquestes, per tant és innecessari fer més càlculs.

Així, el clustering que ens doni el valor més petit de validació, direm que és tracta del valor de K òptim de l'algoritme K-means. Per exemple, per al mapa de metro de Londres que té 11 línies, com podem veure en la figura 2.6, el nombre mínim de clústers és 19.

Un cop ja tenim el nombre de clústers, apliquem de nou el mètode anterior amb la K calculada.

2.2.4.2 Mètode 6. Mesura de la Intra-variança

Aquest mètode explora una altra manera de trobar la K automàticament de manera que ens permeti fer una divisió de la imatge en un nombre de clústers que s'adapti a les necessitats de cada imatge. En el mètode anterior, el que fem és un càlcul intra-cluster i inter-cluster per a obtenir el nombre de clústers que ens permeten fer una bona segmentació de la imatge, però hem vist que el valor calculat, en la majoria dels casos, no s'ajusta al nombre de clústers que cada imatge necessita per a obtenir bons resultats. El valor de K que ens dona moltes vegades és un valor molt gran, de manera que si, per exemple, tenim una imatge amb quatre línies, el mètode ens dona un valor de K de quinze, cosa que ens genera molts clústers amb soroll o, fins i tot, clústers que contenen dos cops la mateixa línia. A més, es fan més càlculs dels realment necessaris amb tants clústers.

Observant els valors intra-cluster i inter-cluster hem pogut observar que quan l'algoritme K-means va afegint més clústers, la diferencia entre els valors intra-cluster,

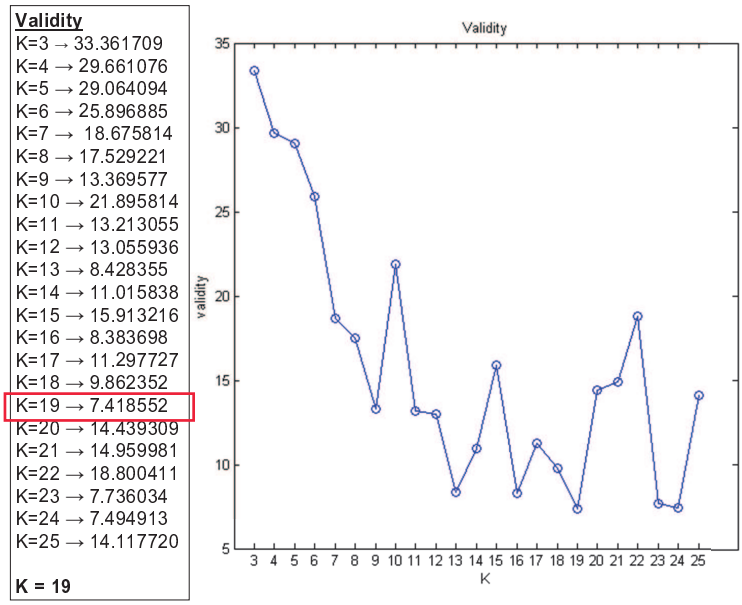


Figura 2.6: **Valors de validació segons el nombre de clústers del mètode 5.** *Tenint en compte que el mapa de Londres té 11 línies, a la gràfica podem veure que el valor més petit correspon a* $K = 19$.

amb respecte el nombre de clústers, és cada cop més petita. Hem pensat que podríem aplicar un threshold a on la diferencia entre aquests valors ja fos suficientment petita i que no variés massa a cada nou clústers afegit, però en algunes ocasions ens trobem que entre diferencies grans, n'hi ha de petites, cosa que ens fa posar el threshold abans d'hora.

A continuació hem volgut veure gràficament aquesta variació del valor intra-cluster i hem pogut observar millor que, efectivament, quan l'algoritme K-means va afegint més clústers arriba un punt en el que la variable intra-cluster s'estabilitza, tal i com es pot veure a la figura 2.7, i hem vist una relació directa entre aquesta estabilitat i el nombre de clústers que necessitem per treure bons resultats al K-means.

Així doncs, hem agafat tots els valors intra-cluster i hem calculat la mitjana, la qual hem agafat com a threshold. Per tant, el nombre de clústers serà el corresponent al valor de K que està just per sobre d'aquest threshold. D'aquesta manera el número de clústers s'ajusta millor a les característiques de cada imatge. Hem comprovat experimentalment que aquest threshold correspon al moment en el que els valors intra-cluster comencen a estabilitzar-se.

Com en els mètodes anteriors, el primer que fem és treure tots els píxels de la imatge original corresponents al fons. De la imatge resultant apliquem l'algoritme K-means amb valors de K entre 3 i 25. De cada K-means calculem el valor intra-cluster i l'emmagatzemem en un vector. Un cop tenim tots aquests valors, fem la mitjana i

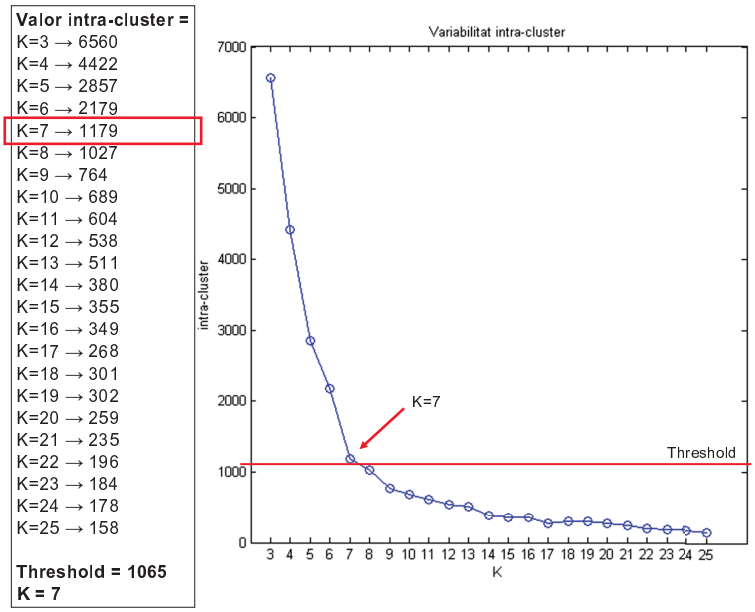


Figura 2.7: **Gràfic de la variabilitat intra-cluster del mètode 6.** *En aquesta imatge podem observar la gràfica de variabilitat intra-cluster del mapa de metro de Rotterdam, que consta de 5 línies. Tal i com indiquen els càlculs de l'esquerra, la $K = 7$, definida pel threshold calculat de la mitjana dels valors intra-cluster.*

ens quedem el valor de K que estigui just per sobre d'aquest threshold. Aquest serà el valor de K que farem servir per a tornar a fer el K-means, el qual ens retornarà K clústers que seràn possibles línies. Cada clúster, es pinta en una imatge binaria, la qual, mitjançant la transformada de Hough, ens dirà si pertany o no a una línia. En el cas que no ho sigui, la descartem. A la figura 2.8 podem veure una millora important entre aquest i el mètode anterior.

2.2.4.3 Mètode 7. Solució ad-hoc sense paràmetres

Observant un mapa de metro qualsevol, sense tenir en compte el fons de la imatge, podríem veure a primera vista quants clústers farien falta per a fer una bona segmentació de les línies. Això és el mateix que dir que sabríem quantes línies hi ha al mapa de metro que estem mirant. Si la imatge conté milers de colors diferents, perquè podem només apreciar uns quants? La resposta és clara, i és perquè hi ha més píxels a la imatge d'uns certs colors que d'altres, els quals fan que la resta, que son simples degradacions dels colors que realment veiem, siguin imperceptibles per al nostre ull.

En aquest mètode fem servir aquesta hipòtesi per a separar les línies dels mapes de metro automàticament. Tal i com veiem a la figura 2.9, si apliquem a la imatge, traient-li el fons, un K-means amb 50 clústers i fem un histograma dels píxels que conté cada clúster, podem observar que hi ha alguns d'aquests que sobresurten de

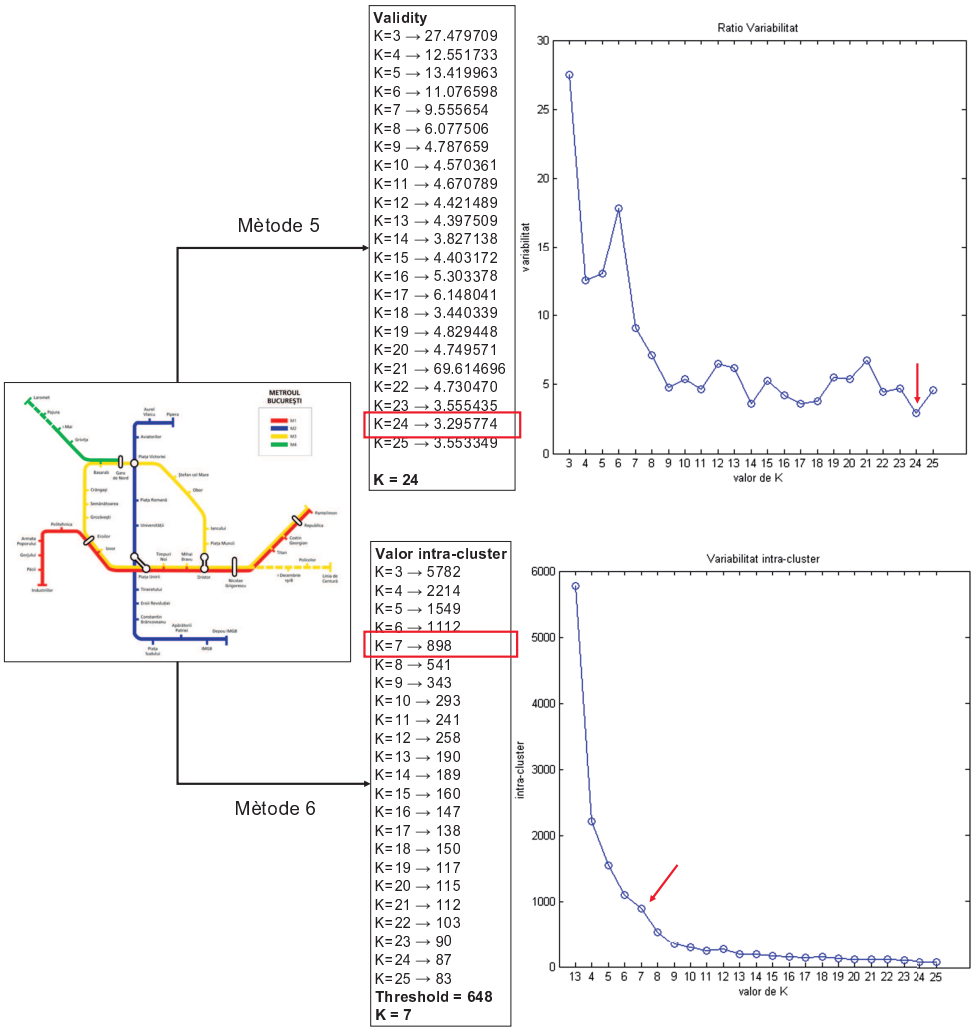


Figura 2.8: **Comparació de la selecció del nombre de clústers per als mètodes 5 i 6 del mapa de metro de Bucarest.** *Aquest mapa de metro conté 4 línies. Es pot veure clarament com el mètode 5 ha trobat com a valor òptim de K un valor massa alt, metre que al mètode 6 hem ajustat aquest valor a un número aproximat al nombre de colors que hi ha al mapa.*

la resta suficient com per poder dir, quasi amb seguretat, que són aquests els que contenen els píxels corresponents a les línies del mapa de metro.

Un cop tenim aquest histograma, ordenem de manera descendent els clústers i apliquem un threshold, el qual calculem segons la diferencia entre ells, de manera que agafem aquells clústers que la distancia al següent sigui més gran que un 2% del valor del clúster amb més píxels. Aquest percentatge l'hem escollit experimentalment per a no tenir massa pèrdua d'informació, buscant un equilibri entre les línies detectades

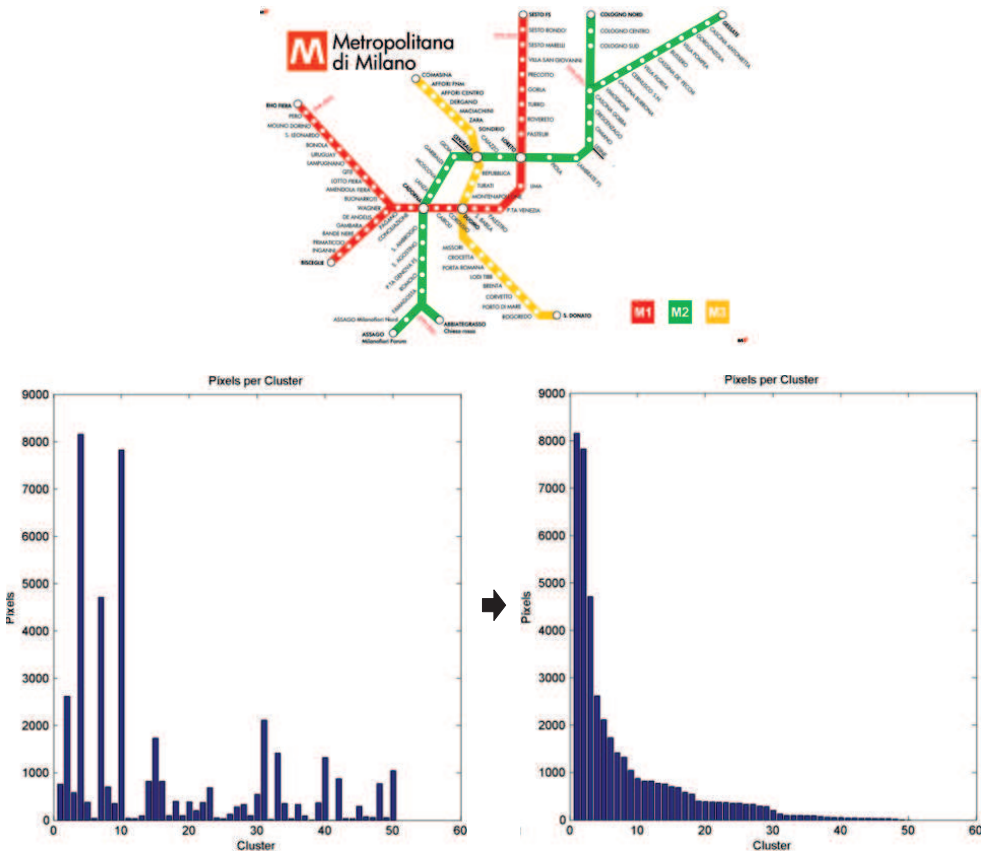


Figura 2.9: **Histograma del mapa de metro de Milà del mètode 7** *Al histograma es veuen clarament tres clústers que contenen més píxels que la resta. La diferència entre ells és veu molt més marcada quan ens ordenem de manera descendent.*

correctament i el soroll generat.

Coneixent ja quins són els clústers que ens interessen, procedim a extreure les imatges resultants.

Hem agafat 50 clústers perquè experimentalment hem comprovat que, quants més clústers hi hagin, més diferència hi ha entre aquells que ens interessen i la resta. Una imatge conté molts píxels i aquests s'han de repartir, a l'hora d'aplicar l'algoritme de clustering, entre els seus clústers. Si entre un grup de píxels hi ha molta similitud, estan molt pròxims entre ells i aquest grup és molt gran, com és el cas d'una línia de metro, el clústers contindrà molts píxels. En canvi, si en un grup de píxels hi ha poca similitud, com és en el cas de la degradació de colors a les voreres del text o les línies, com més clústers hi hagin, més divisions es faran en aquest grup per fer-ne de nous.

A l'estudi realitzat en els mètodes anteriors hem pogut comprovar que és complicat trobar un mètode de selecció del valor de K que funcioni bé. A més, en alguns casos

teníem que la transformada de Hough descarta clústers que contenen línies, deixant passar altres que no. Per aquests motius hem volgut provar aquest mètode, el qual té l'avantatge que no té cap paràmetre i es tracta d'una evolució de tots els mètodes estudiats anteriorment. Tal i com veiem a la figura 2.10, es pot veure com tenim una millor segmentació de les línies.

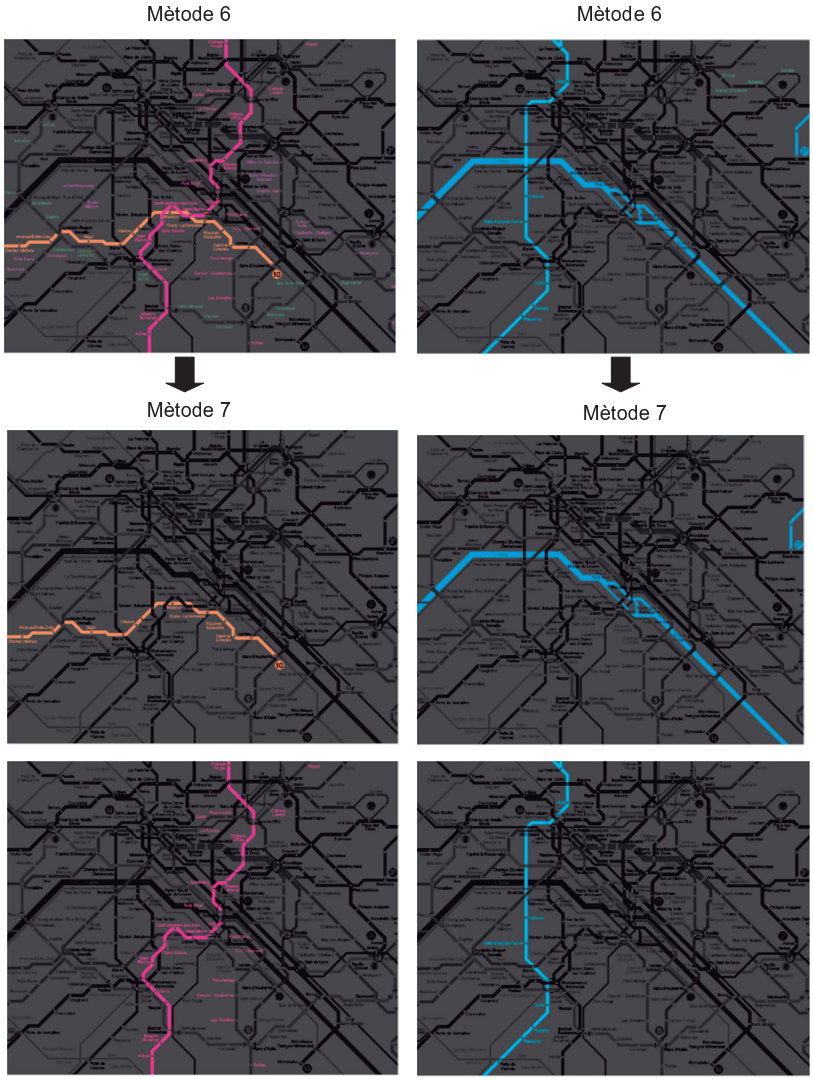


Figura 2.10: **Segmentació de les línies del metro de París amb el mètode 7.** *A les imatges del mètode 6 podem veure una mala segmentació de les línies, mostrant dues en comptes d'una. Per al contrari, veiem com al mètode 7 ens les separa.*

Capítol 3

Resultats experimentals

En aquest capítol descriurem la base de dades que hem fet servir per a fer els experiments. També enumerarem les mètriques utilitzades per a la avaluació del mètodes. A continuació farem avaluacions qualitatives i quantitatives dels resultats obtinguts comparant els mètodes entre ells, explicant els problemes trobats en cadascun d'ells i les solucions a les que hem arribat. Finalment, farem un anàlisi de l'aplicació.

3.1 Base de Dades

En la segmentació de línies de mapes de metro no hi ha cap base de dades feta que puguem fer servir en el nostre projecte. Per tant, hem creat la nostra pròpia a partir d'imatges tretes de la xarxa. Farem servir una base de dades de 21 imatges (veure taula 3.1), que contenen mapes de metro de diferents ciutats del món, de mides entre 474x586 i 1050x1200. Tenint en compte que la majoria dels mapes de metro són molt semblants, doncs en la majoria dels casos hi ha línies de colors sobre un fons blanc, hem intentat que hi hagi diversitat entre les imatges escollides. Aquestes es poden agrupar segons el gruix de les línies, la quantitat d'aquestes al mapa, que pot variar entre 3 i 21, i el color del fons, que en algun cas no es del color blanc que acostumen a ser. A la figura 3.1 podem veure les diferències entre els mapes.

A l'hora d'escollir el format de la imatge teníem, per un costat, imatges amb format GIF, on els colors són purs i treballar amb aquest format no seria realista, i per una altre banda imatges agafades directament de la càmera d'un dispositiu mòbil i que es guardaven en format JPEG, és a dir, comprimit amb pèrdua. En el primer cas, ens trobàvem amb unes imatges massa fàcils de tractar i amb un mètode simple el problema hauria estat solucionat, doncs a la imatge, hi ha tants colors reals com els que es poden veure. En el segon cas, tot i que s'han realitzat proves preliminars, s'ha vist que el problema seria massa complex per als objectius d'aquest estudi. Primer, perquè la qualitat d'una imatge feta per un dispositiu mòbil no és gaire alta; segon perquè aquí entraria en joc la lluminositat i la saturació, a més de la nitidesa. La

Mapa	**Mida**	**Colors**	**Colors reals**	**Línies**	**Comentaris**
Amsterdam	474x586	7	1270	5	LN
Barcelona	1200x605	14	256	12	LP, CS
Bucarest	744x600	7	3291	4	LN
Budapest	783x530	10	579	5	LP, RT
Cairo	700x740	5	5	3	LN
Canal	1200x900	16	1104	15	LN, CS
Delhi	1024x754	7	5024	5	LP, RT
Estocolm	730x632	5	52102	3	LN, RT, FNB
Londres	1200x763	13	25626	11	LP, CS
Hamburg	1200x868	11	13165	7	LP, RT
Kiew	1200x754	7	178	4	LP, RT
LosAngeles	1050x750	11	8783	8	LP, CS
Madrid	854x1200	12	1900	11	LN, CS
Mejico	800x791	13	9370	11	LN, CS
Washington	966x860	10	8261	5	LG, RT
Milan	800x545	6	4346	3	LN
Paris	903x722	23	34	21	LN, CS, RT
Rotterdam	800x546	8	3682	5	LN
SantPetersburg	800x1013	9	13456	5	LP, CS, RT
Tokyo	1200x796	17	2884	14	LN, CS, FNB
WashingtonDC	1050x1200	9	999	5	LN, RT

Taula 3.1: **Base de Dades.** *En aquesta taula tenim les imatges fetes servir. La columna* Colors *representen els diferents colors que l'ull pot veure a la imatge.* Colors reals *són els colors que realment conté la imatge. Y la columna* Línies *són les línies de metro que conté el mapa.*
Comentaris: LN: línies de mida normal; LP: línies primes; LG: línies gruixudes; CS: colors semblants a les línies; RT: fons geogràfic, com un riu, tros de terra, etc.; FNB: Fons d'un altre colors que no és blanc.

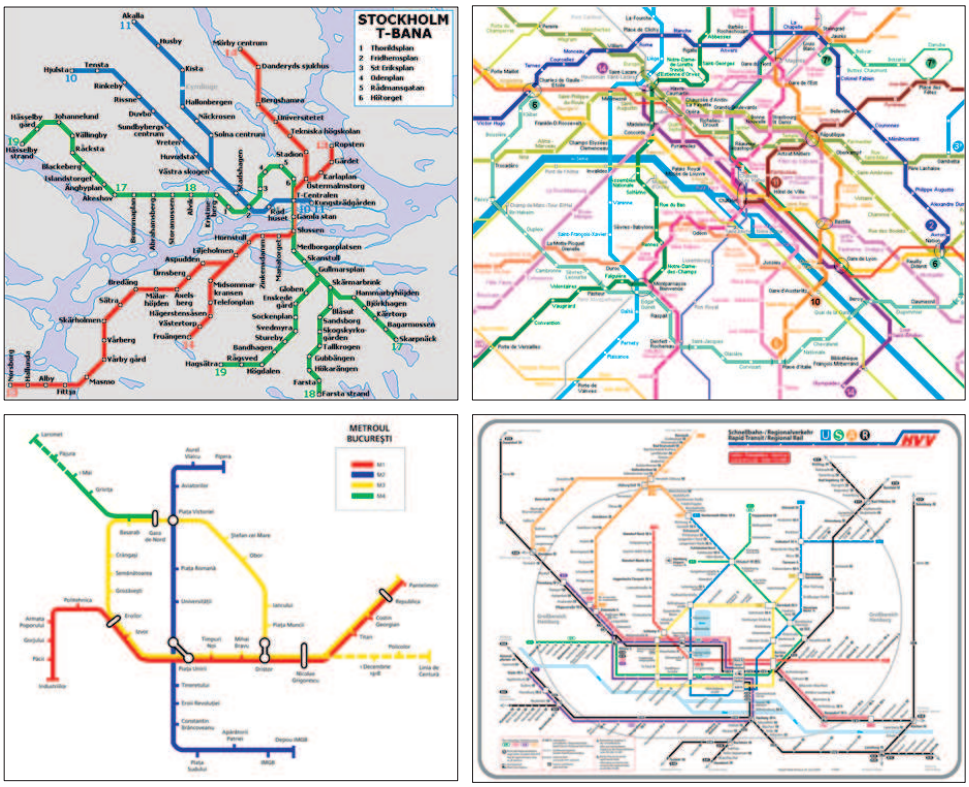


Figura 3.1: **Exemples d'imatges de la base de dades** *La imatge de dalt a l'esquerra representa el mapa de metro d'Estocolm, amb una mica de mapa cartogràfic de fons sense blanc. A la imatge de dalt a la dreta tenim el mapa de metro de París, amb moltes línies i de colors semblants. També es pot veure un riu que es confon amb les línies. La imatge de sota a l'esquerra és un exemple de mapa de metro més comú. Es tracta del mapa de metro de Bucarest. I per últim, la imatge de sota a la dreta correspon al mapa de metro de Londres, amb moltes línies i molt primes.*

imatge s'hauria de tractar abans de fer la segmentació, i a vegades, tot i així, podria no donar una imatge suficientment nítida com per treballar amb ella.

Així doncs, ens vam decantar per un format que estigués dins d'un terme mig. Les imatges en format PNG tenen suficient qualitat com per poder tractar-les sense grans problemes, doncs es tracta d'un format amb compressió sense pèrdua, tot i que així, el problema continua sent complex degut als següents factors:

- El problema del *anti-aliasing*, el qual fa que les voreres d'una línia es difuminin per a evitar canvis bruscos de color. Aquest fenomen provoca un increment del nombre de colors reals de la imatge i complica, en molts casos, la bona segmentació de les línies. A la figura 3.2, es pot veure l'efecte de l'anti-aliasing en una de les imatges que fem servir.

- Treballem amb imatges a diferents escales. El temps de còmput pot variar depenent de la mida de la imatge i hem de trobar mètodes vàlids per a qualsevol mida de la imatge de manera que no impliqui una pèrdua de temps important.
- Hi ha, a més, línies amb diferents mides en diferents mapes de metro, i tractar una línia ben definida no és el mateix que tractar línies molt primes que, sumant el primer factor, la segmentació d'aquestes pot ser complicada.

Figura 3.2: **Anti-aliasing en el mapa de metro de Hamburg.**

3.2 Mètriques d'avaluació

Segons la definició donada pel *IEEE standard glossary of software engineering terminology* [2], una mètrica és una mesura quantitativa del grau que un sistema, component o procés posseeix un atribut donat. Les mètriques es fan servir per a poder fer una avaluació objectiva d'allò que es vol mesurar, analitzar-lo i comprendre millor els seus atributs. També ens dona un control sobre la qualitat del producte de manera que podem fer canvis per intentar millorar-lo. No existeix una mètrica definida que ens permeti mesurar qualsevol sistema, procés o component; i molts cops es poden fer servir més d'una per a un sol procés d'estudi. En el nostre cas es poden fer servir diferents tipus de mètriques, com per exemple calcular el nombre de píxels de cada línia segmentada respecte a la resta de píxels de la imatge, però això comportaria major complexitat al problema, seria molt costós i realment no contindria informació valuosa sobre quin dels mètodes funciona millor (per exemple, no ens importa si una línia no es segmenta en tot el seu gruix).

La mètrica que hem fet servir és la correcta detecció i segmentació de les línies de metro de cada imatge. Així doncs, direm que una imatge està ben segmentada quan ha separat les línies correctament sense tenir en compte el nombre de píxels sinó:

1. Si la línia segmentada va de principi a fi.

2. Si la línia té una amplada mínima per a poder ser vista i identificada fàcilment.

3. Si hi ha o no solapament de línies de manera que impedeixi la identificació.

4. Si el possible soroll que pugui haver a la imatge impedeix o no el correcte seguiment de la línia.

Tindrem com a *false positive* (FP) aquella segmentació de la imatge que no representa una línia, però que s'ha segmentat com a tal. Tractarem com a *false negative* (FN) aquelles línies que no apareixen o han estat descartades en la segmentació. Els *true positive* (TP) seran les línies que han estat ben segmentades. Amb aquestes dades farem un càlcul de *precision* o PPV (*Positive Predictive Value*) mitjançant l'equació

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

I del *recall*, també anomenat TPR (*True Positive Rate*), amb equació

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

La mesura *precision* és la fracció de casos recuperats que son rellevants, mentre que el *recall* és la fracció de casos pertinents que es recuperen. Si tenim un alt *recall* significa que l'algoritme retorna la major part de resultats rellevants, mentre que una alt valor de *precision* significa que l'algoritme retorna resultats substancialment més rellevants que irrellevants.

I per acabar, l'ultima mesura que farem servir serà la de *F-measure*, la qual combina les mesures *precision* i *recall*. Es tracta de la mitjana harmònica entre *precision* i *recall*. La seva equació és

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.3}$$

3.3 Avaluació

En aquesta secció farem una comparativa entre els mètodes explicats anteriorment i avaluarem el funcionament de cada mètode segons els resultats obtinguts en els experiments i fent un anàlisi visual i quantitatiu d'aquests. A més, donarem el temps mitjà de càlcul en cada mètode, tenint en compte que els experiments s'han realitzat en MATLAB amb un processador Pentium (R) Dual-Core amb memòria RAM de 4.00GB.

3.3.1 Mètodes amb intervenció

3.3.1.1 Mètode 1. Assignació directa de color

Els resultats obtinguts en aquest mètode són bastant bons, però en algunes línies hem pogut comprovar que l'algoritme no complia amb les expectatives. El principal

problema amb el que ens hem trobat es que en els mapes amb línies molt fines, el valor RGB del píxel seleccionat coincidia amb pocs o cap RGB de la resta de píxels de la imatge de manera que no feia possible una correcta visualització de la línia com indica la figura 3.3.

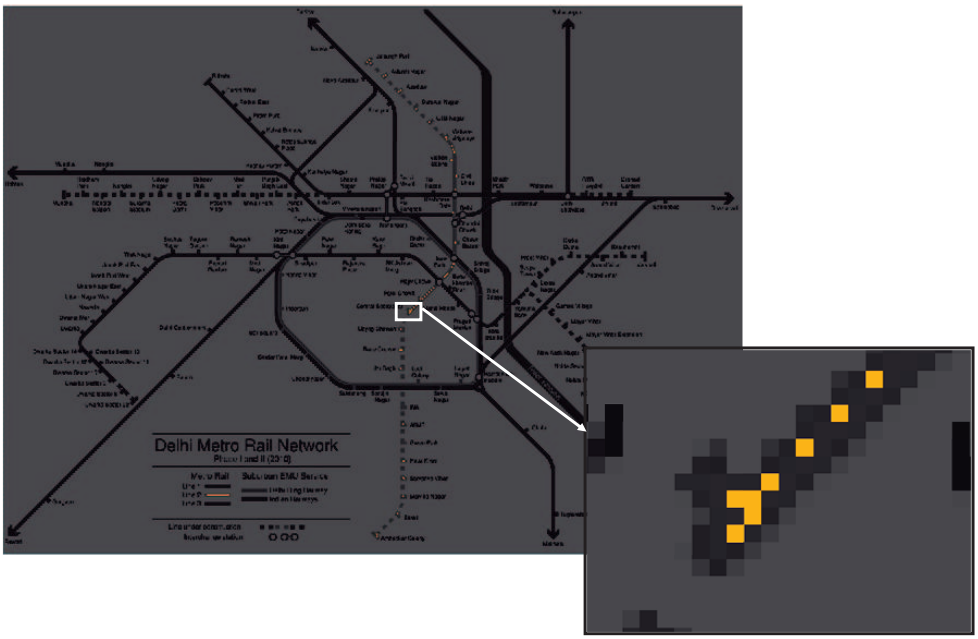


Figura 3.3: **Exemple de línia malament segmentada.** *Aquest és el mapa de metro de la ciutat de Deli. Podem veure com aquest mètode no ha segmentat bé la línia, doncs la degradació del color ho fa impossible.*

A més a més, en el cas de que les imatges tinguin una degradació de color a les voreres de les línies, a la segmentació els contorns surten poc definits i dona com a resultat una imatge visualment poc neta com es pot observar a la figura 3.4.

3.3.1.2 Mètode 2. Assignació aproximada de color

En el segon mètode de segmentació amb intervenció de l'usuari, el qual té la intenció de corregir els problemes del mètode anterior, en totes les imatges que tenen els problemes comentats s'han pogut corregir tal com podem veure al capítol 2 (figura 2.2), però com a conseqüència, ha sorgit un altre problema. En el cas de tenir dues línies de colors molt semblants en una mateixa imatge, aquest rang de valors RGB que s'ha fet servir, provoca que classifiqui totes dues línies com la mateixa. Una confusió que no ajuda a la bona comprensió de la imatge per un daltònic. Aquest problema el podem veure a la figura 3.5. Tot i que es van provar amb altres rangs de RGB, amb els més estrictes continuàvem tenint problemes per a poder definir una línia correctament i, en alguns casos, es segmentaven igualment dues línies per la proximitat de colors.

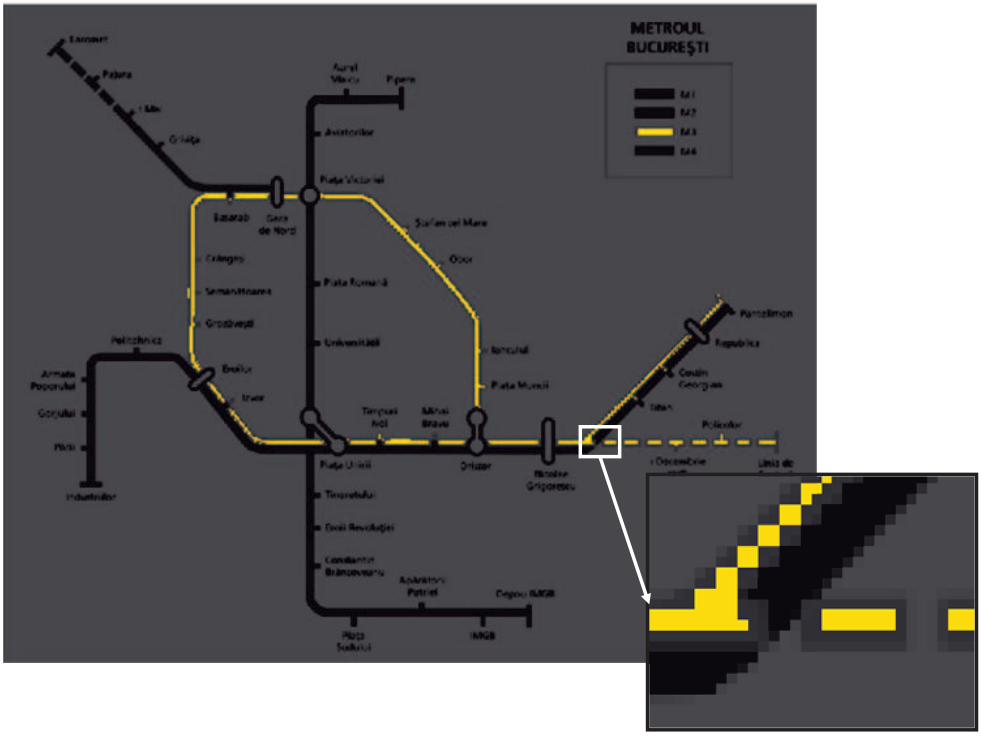


Figura 3.4: **Exemple d'una línia de metro ben segmentada, però amb voreres molt marcades.** *Aquesta imatge és el mapa de metro de Bucarest, on es pot veure com, a les parts de la línia vermella que no son horitzontals o verticals, les voreres es veuen molt pixelades.*

Amb rangs més alts, aquest problema augmentava exponencialment, per tant, vam haver d'agafar un valor mig.

A la taula 3.2 tenim els resultats obtinguts dels dos mètodes anteriors. Podem veure que els resultats dels dos mètodes són molt semblants. Això és perquè, mentre que en un tenim el problema que no ens segmenta bé les línies, en l'altre, a l'hora que ens soluciona aquest problema, ens apareix un altre que fa que es segmentin vàries línies de cop, fent que els resultats d'avaluació siguin semblants. Tot i així, i tenint en compte que ens interessa que es segmentin bé el major nombre de línies possibles, i tal com ens ho indica el valor de *F-measure*, podem dir que el segon mètode és el que segmenta millor.

Els temps de càlcul per a tots dos mètodes és < 1s.

En aquesta secció hem obtingut resultats molt bons, amb un 87% i un 88% de *precision-recall*. Es tracta d'un valor molt alt i assegura que la part de la interfície amb intervenció de l'usuari sigui útil per als daltònics.

(a) (b)

Figura 3.5: **Exemple del problema sorgit al segon mètode.** *A la imatge (a) tenim la segmentació del primer mètode d'una línia de metro. A la imatge (b) podem veure com amb el segon mètode, no només ens mostra aquesta línia, sinó que també ens mostra línies de colors semblants a la seleccionada.*

	TP	**FP**	**FN**	**Precision**	**Recall**	**F-measure**
M1	140	22	22	0.86	0.86	0.86
M2	142	21	20	0.87	0.88	0.87

Taula 3.2: **Taula de resultats dels mètodes amb itervenció de l'usuari**

3.3.2 Mètodes sense intervenció

3.3.2.1 Mètode 3. L'espai dels noms de color

En el primer mètode de la segmentació sense intervenció de l'usuari, apliquem Color Naming abans de l'algoritme de clustering per a poder tenir una imatge amb colors més purs, sense degradació de color. D'aquesta manera intentem desfer-nos dels problemes amb els clústers amb soroll del mètode anterior. Aquest mètode, a més, ens serviria per a un etiquetat de les línies amb el nom dels colors per a una posterior aplicació. Com podem veure a la figura 3.6, la segmentació de moltes de les línies és correcta, separant-les en diferents clústers i amb una bona definició a conseqüència de l'assignació del mateix valor RGB, per mitjà de la funció Color Naming, als píxels corresponents a cada línia.

Encara que apliquem Color Naming per evitar l'*anti-aliasing*, fer-ho ens comporta un problema. En alguns casos, tal i com podem veure a la figura 3.6, dos o més línies de colors semblants els pinta com un de sol. El motiu és que el nombre de colors bàsics és molt limitat i la seva assignació a objectes d'una imatge no pot ser suficient si tenim més tipus de colors que colors bàsics. Es podria ampliar el nombre de colors bàsics, però el fet de fer servir una paleta gran de colors fixa fa que molts colors que es necessitin poden no estar disponible, i molts dels colors disponibles poden no ser

necessaris. Per tant, continuaríem tenint aquest problema.

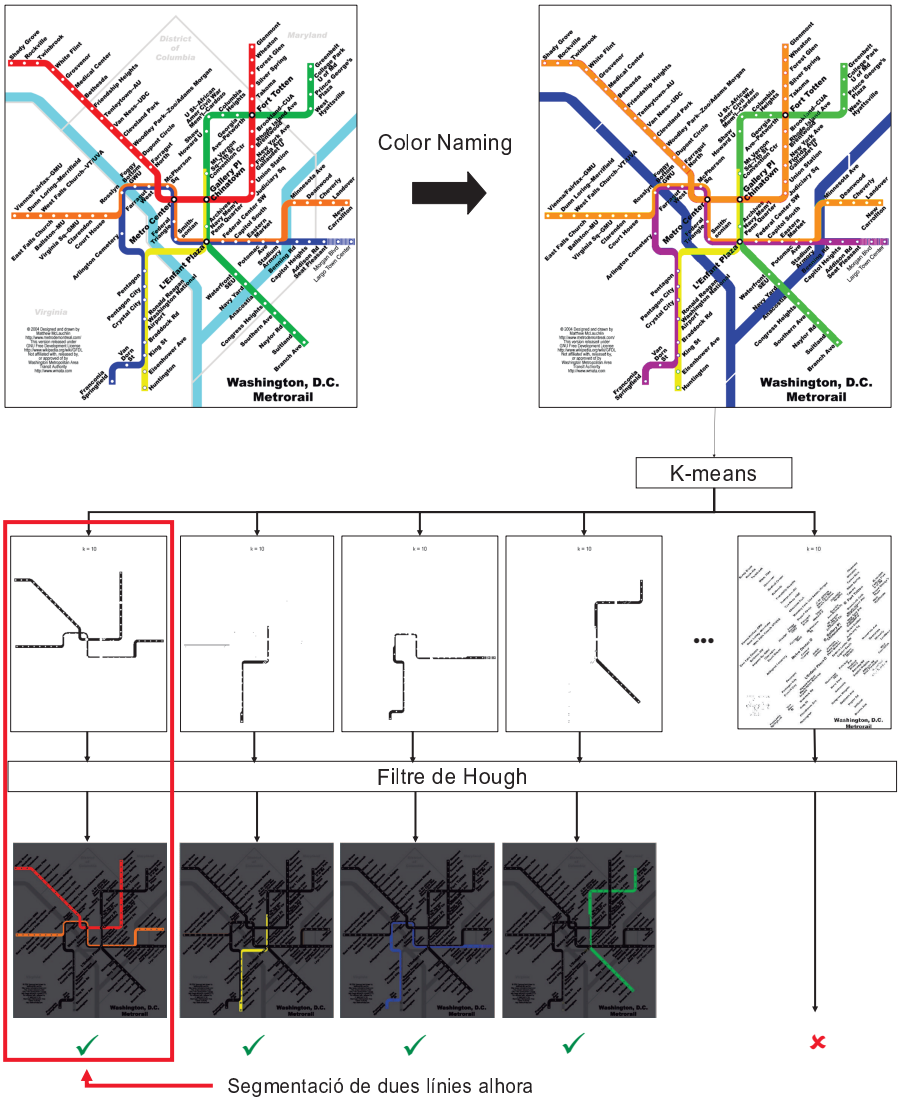


Figura 3.6: **Segmentació amb el mètode 4 del mapa de metro de Washington.** *El mapa de metro consta de 5 línies. Al aplicar un K-means amb 10 clústers, la transformada de Hough descarta aquelles que contenen soroll. Com que la funció Color Naming ens ha retornat la imatge amb dues línies diferents amb el mateix color les tenim agrupades en un mateix clúster, cosa que fa que a la imatge resultant tinguem les dues línies mostrades alhora.*

A la taula 3.3 podem veure que aplicant aquest mètode tenim menys línies ben segmentades que el mètode anterior i amb un valor mes gran de *true positive*, doncs al tenir dues línies el mateix color, son representades pel mateix clústers i això fa que a la

segmentació tinguem dues línies diferents en una mateixa imatge, per tant, dins de la imatge tindrem un *true positive* per la línia ben segmentada, un *false positive* perquè a la mateixa imatge surt una línia que no correspon a la esperada, i un *false negative* perquè aquesta línia malament segmentada ja no la podem classificar correctament, doncs no tenim una imatge amb només aquesta línia.

Els temps de càlcul mitjà esta al voltant de 1 minut i 30 segons.

3.3.2.2 Mètode 4. L'espai RGB

En aquest mètode el que fem és aplicar l'algoritme K-means a la imatge amb un nombre de clústers definit prèviament (el mateix per a totes les imatges), després fem servir la transformada de Hough per a decidir quins dels clústers contenen o no línies.

En molts casos amb aquest mètode obtenim bons resultats. Tal i com podem veure a la figura 3.7, la majoria de les línies estan ben segmentades i n'hi ha poques que quedin sense segmentar. A més, les línies estan ben definides. La transformada de Hough en alguns casos fa un filtrat correcte dels clústers descartant aquelles imatges que contenen text o soroll.

Tot i no funcionar del tot malament, hem vist que la inicialització manual i invariant del nombre de clústers dona problemes. El valor que fem servir, definida al capítol 2, pot anar bé per a aquelles que tenen un nombre aproximat de colors (i quan parlem de colors parlem de colors que es poden veure a la imatge) al nombre de clústers, però per a la resta. A les imatges que tenen menys colors que clústers, l'algoritme K-means genera clústers que contenen voreres de les línies, text o soroll. En el cas de tenir imatges que tenen més colors que clústers, com es pot veure a la figura 3.7, se'ns agrupen diferents colors, semblants entre ells, en un mateix clústers.

Al aplicar la transformada de Hough per a desfer-nos dels clústers que contenen soroll i text, en molts casos ens trobem que no les filtra, doncs hem d'agafar un threshold poc restrictiu per a que els mapes que continguin més línies es puguin filtrar, però suficient com per a no tenir masses imatges amb soroll als mapes amb poques línies. A més, com ja vam explicar al capítol 2, Hough no funciona correctament si hi ha molt de soroll a la imatge. Es podria buscar una solució a aquest problema com comprovar la proximitat dels píxels d'un clústers amb els píxels d'un altre per veure si es tracta de voreres de línies i ajuntar-los si fos el cas, però aquest mètode seria molt costós per haver de fer comparacions píxel a píxel més d'un cop.

Els temps de càlcul mitjà és del voltant de 1 minut.

3.3.2.3 Mètode 5. Mesura de la *Validity*

Aquest mètode busca una K idònia per a cada imatge. Primer busquem el nombre de clústers que ens dona la relació que hi ha entre intra-cluster i la inter-cluster per després aplicar de nou el K-means amb la K calculada.

L'avantatge d'aquest mètode és que no tenim el mateix valor de K, predefinida per a totes les imatges, sinó una calculada específicament per a cadascuna d'elles. Així,

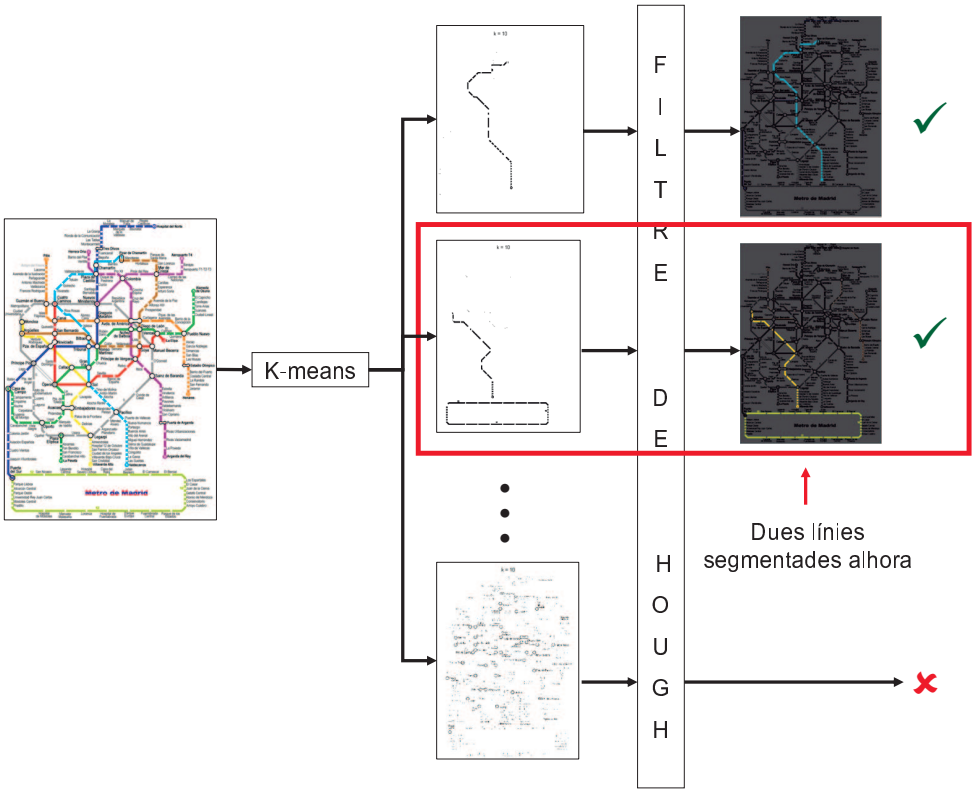


Figura 3.7: **Segmentació amb el mètode 3 del mapa de metro de Madrid.** *El mapa de metro consta d'11 línies. Al aplicar un K-means amb 10 clústers, la transformada de Hough descarta aquelles que contenen soroll. Com que tenim menys clústers que línies, es pot veure que hi ha línies agrupades en un mateix clúster de manera que obtenim imatges amb dues línies en comptes d'una.*

per a imatges amb moltes línies podem tenir més de 10 clústers que puguin abastar totes les línies, metre que si tenim menys de 10, podem ajustar el nombre de clústers a la quantitat de línies. Tal i com podem veure a la figura 3.8, el nombre de clústers calculat en aquest mapa s'ajusta als colors de la imatge.

A la taula 3.3, podem veure com amb aquest mètode hem augmentat el número de línies detectades, però a costa de guanyar més *false positive* de manera que fa baixar molt el *precision*. Per tant, de totes les segmentacions fetes, només un 47% són realment línies.

Com a resultat hem pogut provar, experimentalment, que la K que ens dona, no sempre és la més idònia. Doncs en la majoria de les imatges el número de clústers calculat prèviament és massa gran (veure figura 3.9). Això provoca que es generin clústers amb molta informació que no ens interessa i augmenta la probabilitat de que Hough no les descarti i obtenir un alt nombre de *false positive*. Per tant, en la majoria

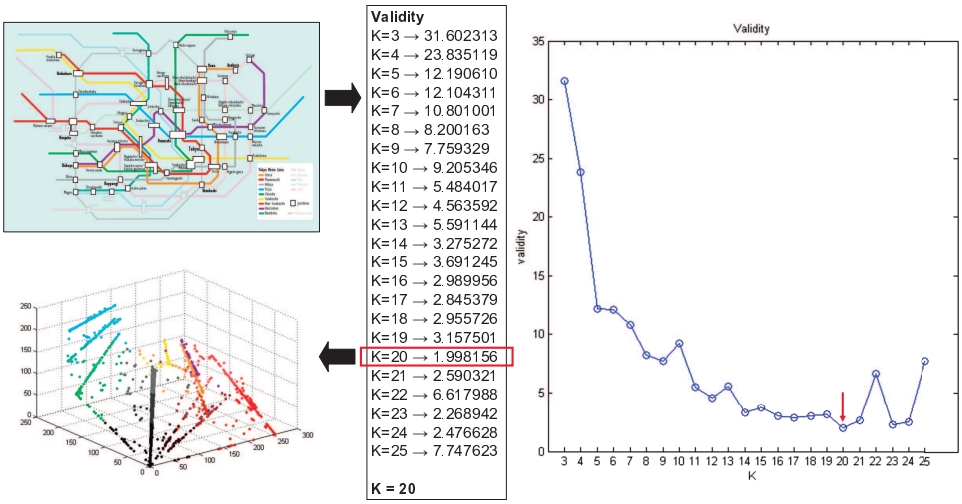


Figura 3.8: **Càlcul del nombre de clústers per al mapa de metro de Tokio.** *Aquest mapa de metro consta de 14 línies. Podem veure com el càlcul que fem en el mètode 5 ens diu que la K òptima és 20, la qual s'ajusta una mica al nombre de línies. Podem veure la distribució d'aquests clústers en el gàfic de l'esquerra.*

dels casos ens trobem amb els mateixos problemes que al mètode anterior.

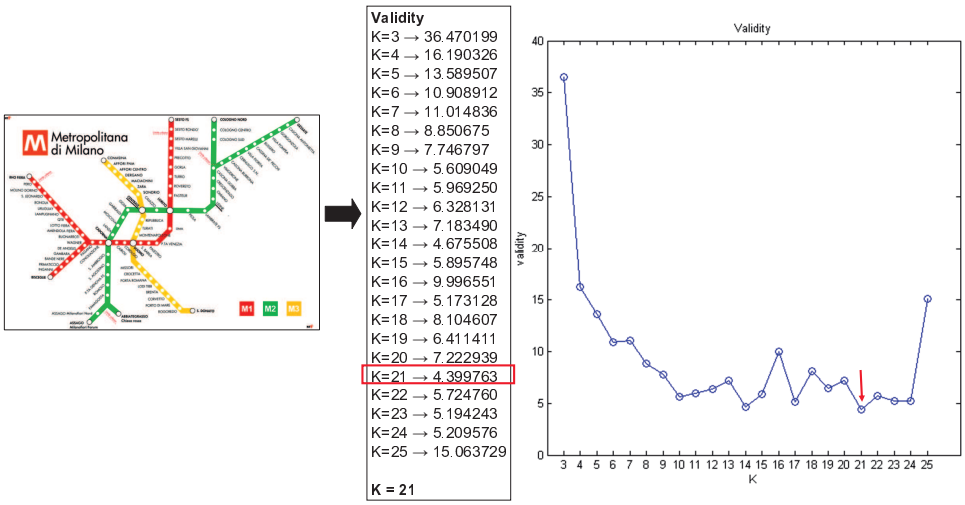


Figura 3.9: **Càlcul del nombre de clústers per al mapa de metro de Milà amb el mètode 5.** *Aquest mapa de metro consta de 3 línies. Podem veure com el càlcul que fem en el mètode 5 ens diu que la K òptima és 21. Aquest valor és massa alt, el qual genera molts clústers amb soroll.*

Els temps de càlcul mitjà esta al voltant de 5 minuts.

3.3.2.4 Mètode 6. Mesura de la intra-variança

En aquest mètode, el que fem és fer un càlcul de K mitjançant la mitjana de la variació de la intra-variança a mesura que augmentem el nombre de clústers. Després, apliquem K-means amb el valor de K calculat i la transformada de Hough per a filtrar els resultats.

Amb aquest sistema podem ajustar millor el nombre de clústers que es necessiten per a cada imatge. El problema del mètode anterior és que en imatges amb poques línies, la majoria de vegades obtenim un nombre de clústers molt elevat. A conseqüència d'això, es fan càlculs innecessaris i s'obtenen molts clústers que poden ser descartables. Amb el càlcul del valor de K que fem en aquets mètode podem trobar un que sigui aproximat al nombre de línies que tenim a la imatge, de manera que la segmentació sigui més ajustada. A la figura 3.10 podem veure com, en el mateix mapa de metro que el de la figura 3.9, el càlcul del nombre de clústers d'aquest mètode s'ajusta molt millor al del mètode anterior.

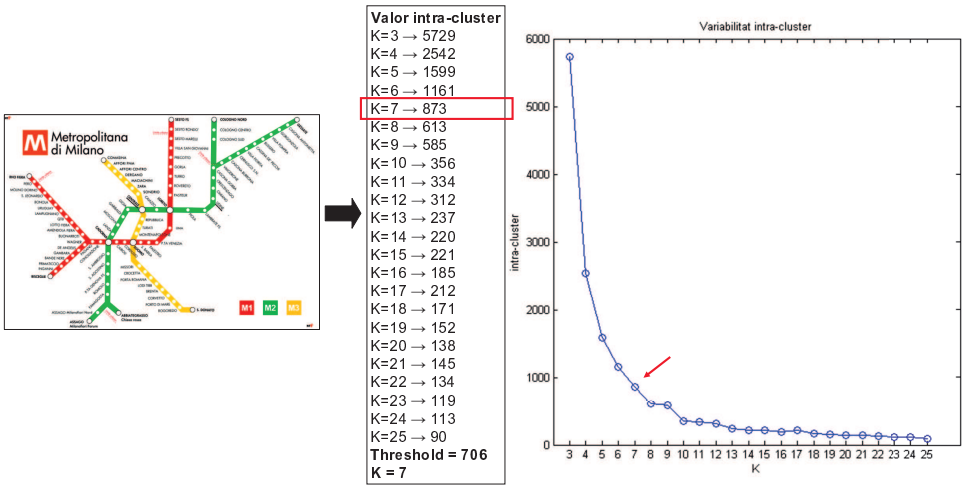


Figura 3.10: **Càlcul del nombre de clústers per al mapa de metro de Milà amb el mètode 6.** *Aquest mapa de metro consta de 3 línies. Podem veure com el càlcul que fem en el mètode 6 ens diu que la K òptima és 7. Comprovem que aquest valor s'ajusta millora a les característiques de la imatge que el mètode 5.*

A la taula 3.3 podem veure com ha disminuït, de manera poc significativa, el nombre de línies ben segmentades i ha augmentat el de línies no detectades respecte al mètode anterior. Però tenim que el nombre de *false positive* ha baixat més de la meitat. Per tant, el valor de *precision* ha augmentat, cosa que fa que ara tinguem que un 64% de les imatges són línies ben segmentades.

Tot i així, el temps de càlcul encara és alt, doncs hem d'executar 22 vegades el K-means per a trobar la K que volem i executar-lo de nou amb aquesta.

Els temps de càlcul mitjà està al voltant de 3 minuts.

3.3.2.5 Mètode 7. Solució ad-hoc sense paràmetres

El que fem en aquest mètode és aplicar a la imatge un K-means amb un nombre de clústers igual a 50. A continuació fem un histograma, ordenat de major a menor, amb el nombre de píxels per clústers. Finalment, i mitjançant un threshold, el qual ja vam explicar al capítol 2, ens quedem amb aquells que continguin més píxels i construïm les imatges corresponents.

Aquest mètode té les següents avantatges:

- El nombre de clústers és sempre 50 independentment del tipus d'imatge que s'hagi de segmentar. Això fa que no tinguem tampoc pèrdua de línies a conseqüencia de tenir un valor petit de K.
- En la majoria dels casos, totes les línies dels mapes de metro contenen molts píxels i, per tant, al histograma hi ha una gran diferència entre els clústers que contenen els píxels de les línies i els que no. Això permet fer una selecció més fàcil.
- El fet d'executar només un cop l'algoritme K-means amb 8 iteracions en comptes de 22 vegades amb 8 iteracions cadascuna en els mètodes anteriors, fa que aquest sigui més ràpid en temps de càlcul.

Podem veure alguns dels resultats obtinguts a les figures 3.11, 3.12 i 3.13.

Un dels problemes d'aquest mètode és que, al tenir tants clústers, les voreres de les línies desapareixen, doncs els píxels corresponents estan en altres clústers a causa de la degradació del color. Tot i així, les línies segmentades són visibles i estan ben segmentades en la majoria dels casos. Un altre problema amb el que ens trobem en unes poques imatges és que el càlcul del threshold que fem, quan tenim clústers amb molts píxels i amb una quantitat similar, talla en aquest punt. Un exemple d'això el trobem a la figura 3.14. Per sort, hi ha molt pocs casos.

A la taula 3.3 podem veure com els resultats d'aquest mètode són molt semblants al mètode anterior i tenint un 0.70 de *recall* i que és més ràpid que tots els anteriors, podem dir que aquest mètode és el que millor funciona.

Els temps de càlcul mitjà està al voltant de 1 minut.

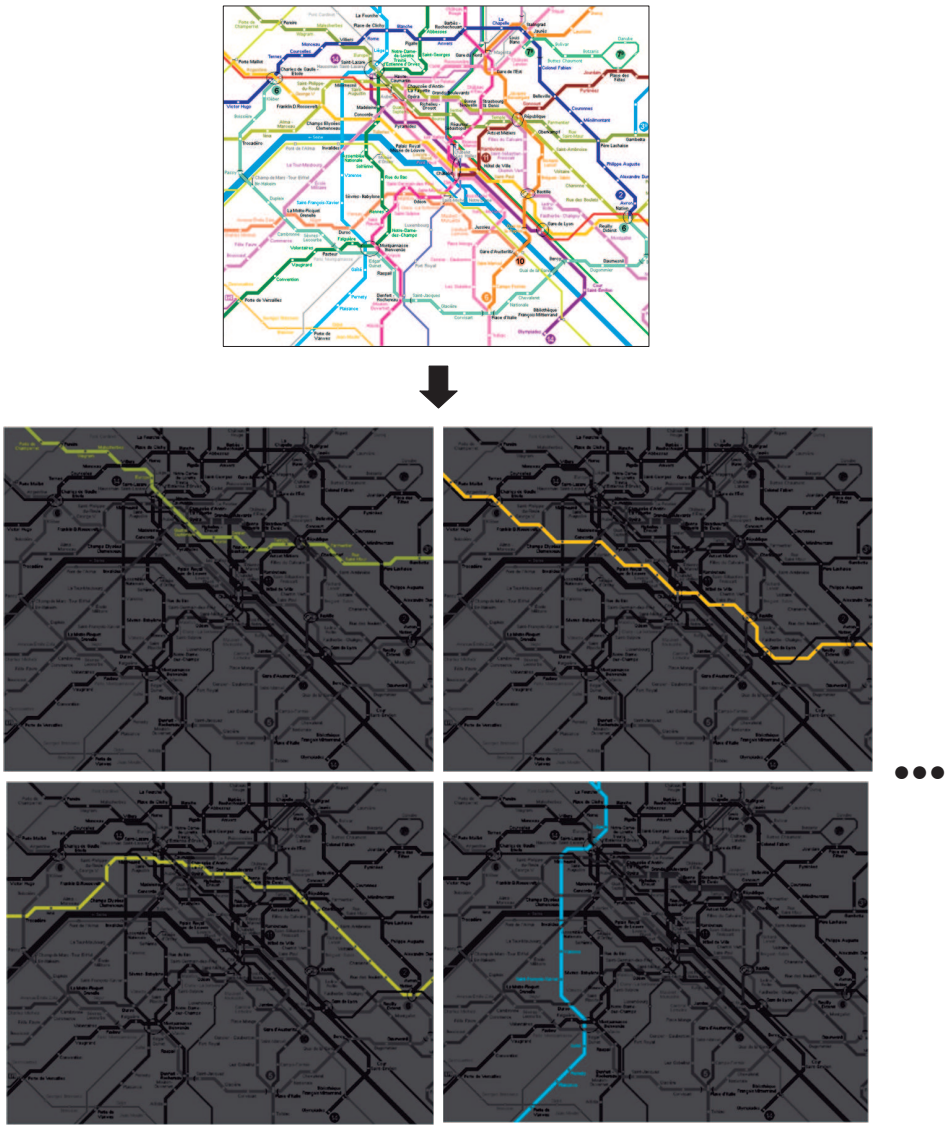


Figura 3.11: **Alguns dels resultats d'aplicar el mètode 7 al mapa de metro de París.** *Com podem veure, en el mètode 7 obtenim una bona segmentació de les línies del mapa de metro de Paris. Podem comprovar-ho amb la figura 3.5 que és un bon exemple de la segmentació de tots els mètodes anteriors.*

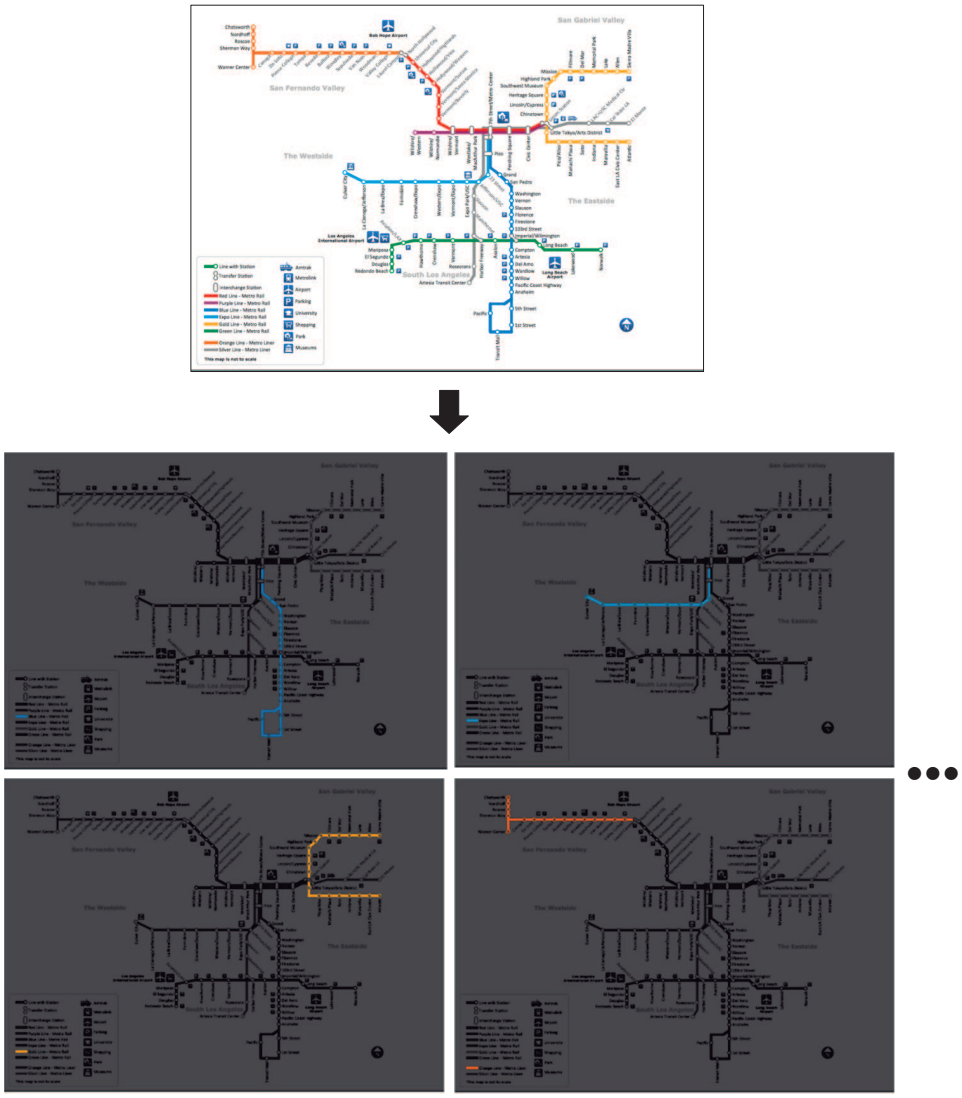


Figura 3.12: **Alguns dels resultats del mètode 7 al mapa de Los Angeles.** *Com podem veure, en molts dels cassos els resultats d'aquest mètode son molt bons.*

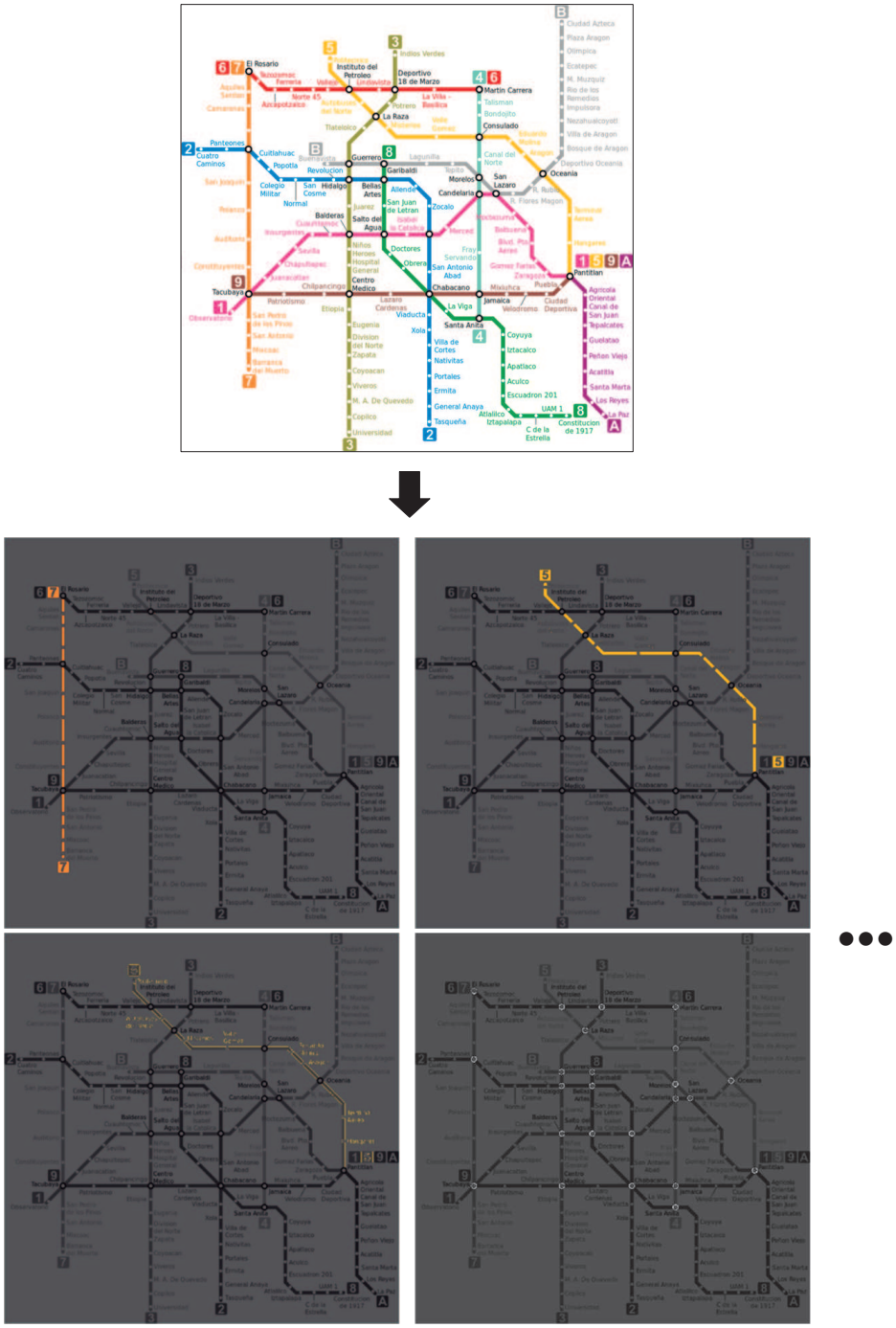


Figura 3.13: **Alguns dels resultats del mètode 7 al mapa de metro de Mèxic.** *Com podem veure, no sempre obtenim bons resultats. En aquest cas, a les dues imatges de sota no tenim cap línia de metro segmentada.*

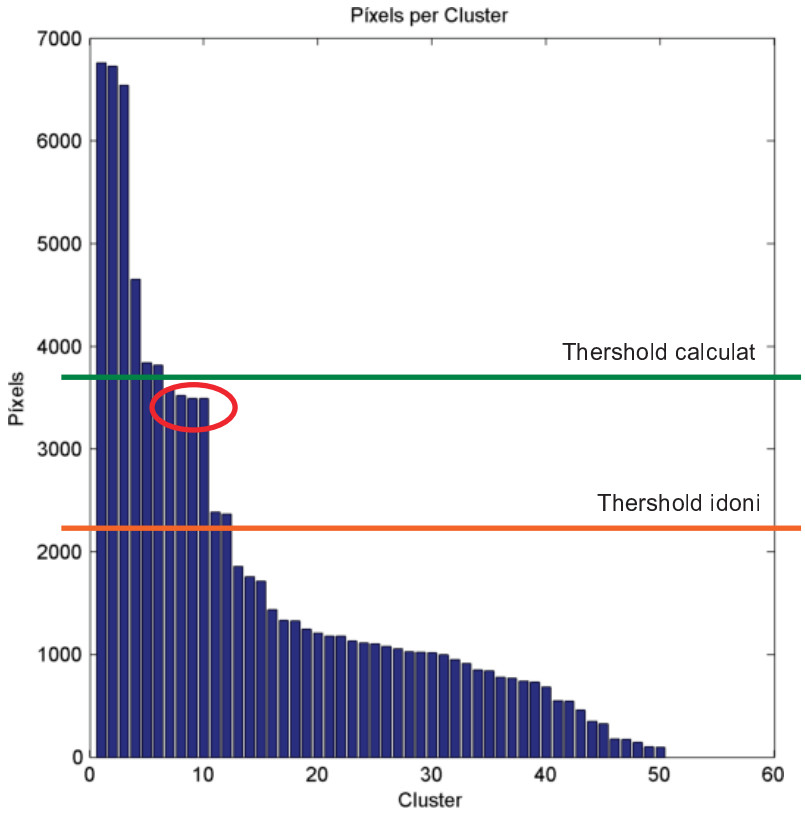


Figura 3.14: **Problema de la selecció del threshold al mètode 7 per al mapa de Londres.** *El mapa de Londres té 11 línies de metro. Com podem veure al histograma, tenim un grup de clústers amb molts píxels, i tots amb una quantitat semblant. Aquest és el motiu per el qual el threshold està per sobre d'ells, quan en realitat hauria d'estar més avall.*

	TP	**FP**	**FN**	**Precision**	**Recall**	**F-measure**
M3	110	97	51	0.53	0.68	0.60
M4	113	81	48	0.58	0.70	0.64
M5	128	145	34	0.47	0.80	0.60
M6	114	63	48	0.64	0.70	0.67
M7	113	60	49	0.65	0.70	0.67

Taula 3.3: **Taula de resultats dels mètodes sense interacció amb l'usuari**

Capítol 4

Aplicació

Un dels objectius d'aquest projecte és l'estudi d'una interfície per a una posterior utilització en un dispositiu mòbil. En aquest capítol explicarem el disseny, el qual podem veure en la figura 4.1, que hem fet i les diferents interfícies que tenim.

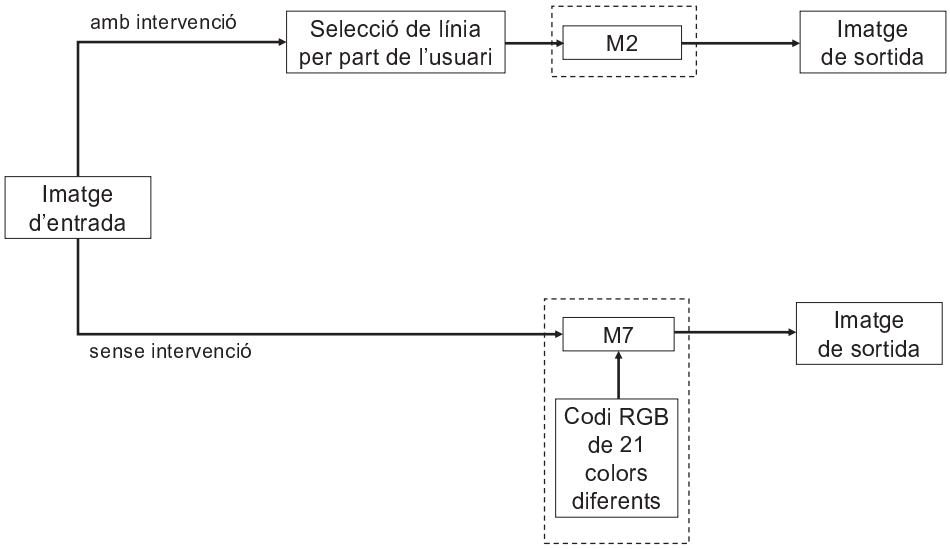


Figura 4.1: **Disseny general dels diferents tipus d'interfície.**

4.1 Amb intervenció

Com ja comentavem al capítol 2, l'aplicació tindrà una petita part d'intervenció de l'usuari, qui tindrà l'opció de fer una selecció de la línia que vol veure i l'aplicació li mostrarà la imatge. A la figura 4.2 podem veure un exemple d'interfície.

L'usuari seleccionarà en la imatge una línia que vol que es mostri de manera que

li sigui comprensible. L'aplicació agafarà d'aquesta selecció el color de la imatge. Posarà la imatge en gris per no perdre informació i mostrarà la línia seleccionada en el seu color corresponent mitjançant el mètode 2 explicat al capítol 2.

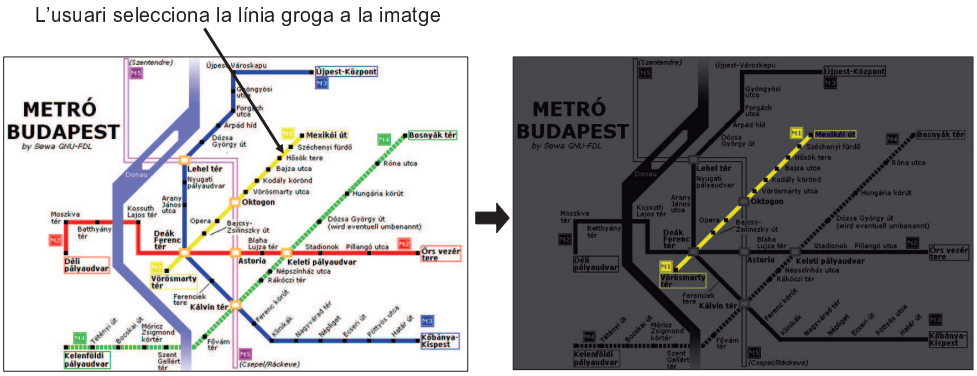


Figura 4.2: **Funcionament de la interfície amb intervenció de l'usuari.**

4.2 Sense intervenció

Al capítol 2 hem vist diferents tipus de mètodes per a fer la segmentació de línies de metro. Agafarem el mètode que funciona millor, que és el mètode 7, per a fer aquesta segmentació de les línies. Tenim, a part, un arxiu que conté una paleta de 21 colors, el quals es mostren a la figura 4.3, escollits per part d'un usuari daltònic amb protanòpsia i deuteranomalia, segons ens indica un test realitzat a la web [1]. Colors que pot distingir més o menys bé.

Figura 4.3: **Funcionament de la interfície sense intervenció de l'usuari.**

El que fem és aplicar el mètode 7 per a la segmentació. Un cop tenim les línies segmentades, posem la imatge original a escala de grisos i assignem un color diferent a cada clústers de la llista de valors RGB que li passem. Així, la imatge de sortida

contindrà una nova imatge amb colors suficientment diferents com per a que l'usuari pugui distingir-los tots.

Hem escollit 21 colors i no més perquè creiem que eren suficients. A més, en la nostra base de dades només tenim, com a màxim, 21 línies. A més, quants més colors tinguem a la llista, més probable és que obtinguem colors que l'usuari no pot diferenciar. Els colors que són més diferents estan al principi, de manera que siguin molt fàcils de diferenciar si tenim poques línies.

En aquest cas només fem servir la paleta de colors per a un sol tipus de daltonisme. A l'aplicació final hauríem de tenir una paleta especialitzada per a cada tipus, de manera que l'usuari pogués seleccionar el seu tipus i fer servir la paleta corresponent.

Un exemple del funcionament el podem veure a la figura 4.4

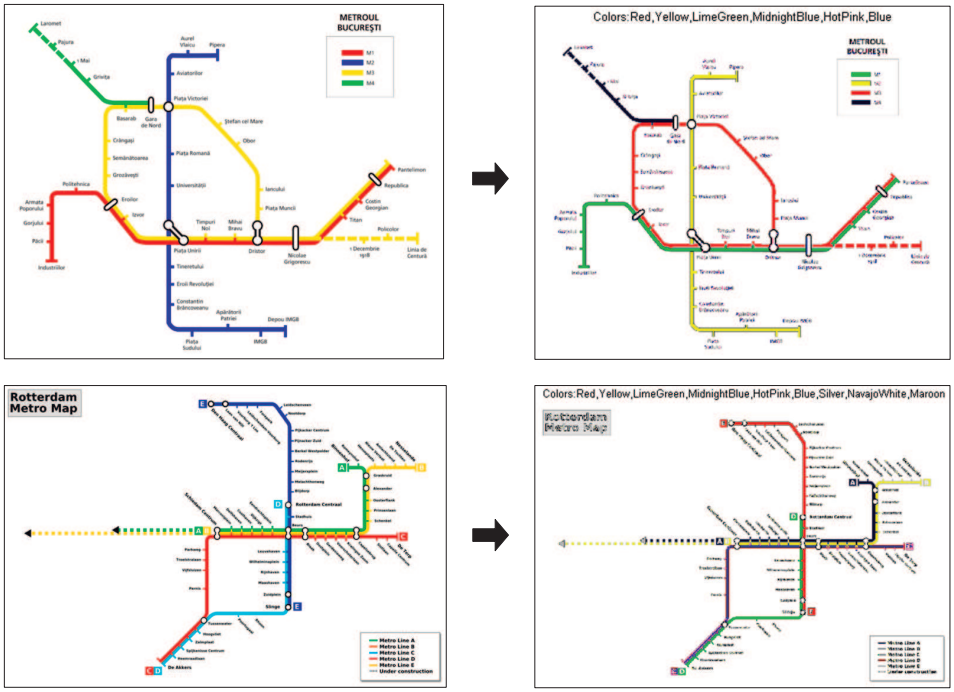


Figura 4.4: **Funcionament de la interfície sense intervenció de l'usuari.**

Capítol 5

Conclusions i línies de continuació

Fer una correcta segmentació i extracció de les línies de mapa de metro, tot i que sembla una tasca fàcil, es pot convertir en un problema molt difícil de solucionar. Els algoritmes han de tractar amb un número variable de colors i s'han de tenir en compte altres factors en les imatges com l'*anti-aliasing*, el número de línies, el seu gruix o els fons de color diferent al blanc.

En aquest projecte s'ha estudiat una aplicació que ajudi als daltònics en la comprensió de mapes de metro basats en colors. Per a dur a terme aquesta tasca, s'ha fet un estudi de la comprensió automàtica dels mapes i s'han desenvolupat dues versions. La primera d'elles amb intervenció de l'usuari i la segona sense. A més, s'ha fet una proposta dels mòduls que aquesta aplicació d'ajuda hauria de tenir, centrant-nos en dos d'ells: la segmentació de color i l'extracció de les línies dels mapes de metro. Dins de la segmentació de color hem analitzat l'ús del K-means com a algoritme bàsic i tots els problemes que es plantegen en la selecció dels seus paràmetres. S'ha explorat, a més de l'espai de color RGB, la hipòtesi de fer servir els noms de color com a descriptor bàsic per a la segmentació, encara que els resultats no han estat els que s'esperaven. Al mòdul de l'extracció de línies s'ha estudiat l'ús de la transformada de Hough, la qual fa una selecció dels resultats obtinguts de la segmentació amb l'objectiu de fer un refinament de les línies. Finalment, s'ha proposat un algoritme basat en inferència heurística sobre el resultat del K-means amb paràmetres prefixats i l'histograma dels segments extrets. Amb aquest mètode podem fer una bona segmentació de colors en un temps raonable per a poder després extreure una imatge que l'usuari pugui comprendre.

Amb la realització d'aquest projecte hem après a crear un prototipus per a la resolució de problemes de visió per computador. A més, a falta d'una base de dades predefinida, hem hagut de crear una nosaltres mateixos tot buscant varietat i una certa complexitat per al posterior estudi. Per a fer l'avaluació del prototipus, hem

realitzat experiments amb els que hem obtingut resultats per a cadascun dels mètodes aplicats en aquest estudi. Hem après a aplicar mesures d'avaluació del rendiment en cada un dels experiments i a fer un anàlisi dels resultats de manera que ens ha permès anar buscant una millora del prototipus. Finalment, hem après a fer una recopilació de tot el treball realitzat i de tots els resultats obtinguts i fer l'escriptura d'una memòria que permeti entendre tot el que s'ha desenvolupat durant el projecte.

Tot aquest treball previ ens ha permès veure la dificultat del problema i em pogut valorar diverses vies de continuació que s'haurien de dur a terme per arribar a implementar una aplicació d'aquest estil, les quals es poden resumir en els següents punts:

- Introduir un mòdul de pre-processament de la imatge per a millorar la seva representació del color, introduint tècniques que permetessin eliminar els efectes d'il·luminació i de les ombres que poden aparèixer en una imatge adquirida en un context urbà i amb un telèfon mòbil qualsevol.
- Fer un estudi sobre les necessitats de cada possible usuari i així definir posteriorment una interfície adaptada a cadascun d'ells i al seu tipus específic de daltonisme.
- Millorar els algorismes desenvolupats i aprofundir més en l'estudi del descriptor de nom de color estenent-lo per a una millor adequació al problema.

Bibliografia

[1] Opticien-lentilles. http://www.opticien-lentilles.com/daltonien_beta/nueva_test_daltoniano.php (Disponible: 14/06/2013).

[2] ANSI/IEEE. *Standard Glossary of Software Engineering Terminology, STD-729-1991*. ANSI/IEEE, 1991.

[3] A.Tukusi. Chromatic glass. http://asada.tukusi.ne.jp/chromaticglass/e/ (Disponible: 14/06/2013).

[4] R. Benavente, M. Vanrell, and R. Baldrich. Parametric fuzzy sets for automatic color naming. *Journal of the Optical Society of America A*, 25(10):2582–2593, Oct 2008.

[5] B. Berlin and P. Kay. *Basic Color Terms: Their Universality and Evolution*. University of California Press, 1969.

[6] R.O. Duda and P.E. Hart. *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Communications of the ACM, 1972.

[7] Bradley C. Grimm. Colorblind vision. https://play.google.com/store/apps/details?id=com.givewaygames.colorblind_ads (Disponible: 14/06/2013).

[8] P.V. HOUGH. Method and means for recognizing complex patterns. In *U.S. Patent 3069654*, 1962.

[9] D. Kaminsky. Dan kaminsky's blog. http://dankaminsky.com/2010/12/15/dankam/ (Disponible: 14/06/2013).

[10] David J.C. MacKay. *Information Theory, Inference and Learning Algorithm*. Cambridge University Press, 2003.

[11] Siddheswar Ray and Rose H. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In *International Conference on Advances in Pattern Recognition and Digital Techniques*, pages 137–143, 1999.