

TESI DOCTORAL

Autor:

Joan Gerard Camarena Estruch

Directors:

Dr. Valentín Gregori Gregori

Dr. Almanzor Sapena Piera

Aplicació de mètriques *fuzzy*
en la millora computacional
d'algorismes de filtratge d'imatges
en color

10 de febrer de 2009

*Amb il·lusió especialment per a Leila i els qui
vindran...*

Agraïments

En primer lloc voldria agrair a la meua família: a la meua dona Àngela, a qui estimo cada dia més. Ella ha sigut eixe pilar que està falcant el cinquanta per cent de la meua vida, sobre el que estem formant el cent per cent d'eixes noves vides; simplement gràcies per ser com eres. Als meus pares que són l'espill on m'he mirat dia a dia; la meua germana i cunyat i les perles d'Anna i Andrea, que els he vist créixer com a família i són un gran model a seguir; Miguel, M^a Ángeles i Vanessa, que m'han acollit a sa casa com si fos un fill més. A tots, gràcies altre cop pel seu recolzament, amor i confiança que sempre m'han oferit. Sempre heu estat ahí, en els moments bons i en els no tan bons, i heu demostrat que sou el millor que una persona pot tindre.

Seguidament estan tot el gruix paquet de persones que s'han encarregat de la meua formació, cosa complicada el mencionar-les a totes les que han exercit sobre mi eixa influència positiva i han aconseguit entre totes fer de mi el qui he arribat a ser. Els primers mestres a qui estic agraït són a Juanito i Trini, els millors professors que he tingut, que m'han impartit assignatures tan importants com respecte, convivència i sentit comú, i m'han esculpit dia rere dia. Comentar als professors que varen despertar en mi les llavors del dubte, del descobrir, del "anar més enllà", ja des de la secundària fins la facultat, José Maria (per ser tan obert), José (per fer dels autòmats màquines perfectes), Valentí (per fer de l'abstracte lo comprensible), Àlvar, Vicent i molts més que no caben ací. Gràcies especialment al Dr. Valentín Gregori Gregori, per la seua confiança i paciència com a director d'aquesta Tesis Doctoral, amb la que culmina la meua formació.

En tercer lloc, agrair a les persones que dins del món universitari han sabut ajudar-me i fer-me mes pla aquest camí llarg i complicat: Alman, Samuel, Jordi, Bernardino, Vicent, Alfred i un llarg etc.

Seguidament anomenar a la gent que han fet que la vida fora del ambient universitari fora més divertit, Alman, Pep, Jordi E., Marc, Jordi B. (amb les respectives, clar està); Vicent, Cristià, Emili, Pablo, Raul, Oscar ; A tota la tropa de l'AMSCA per eixos "Moments de Festa", a la gent de Xeresa que m'ha rebut en els braços oberts: Josep, Melisa, Toni, M^a Jose . . . ;

Per finalitzar, encara que ja anomenats i no per això menys importants, per moltes més raons que ones té la mar, per tota la seua estima, amor, confiança, matinades de treball, arroves de taronja collides, sopars servits, hores de dedicació i moltes més, eternament agraït a Trini i Juanito.

Abstract

Image filtering is an essential image processing task for almost every computer vision system where images are used for automatic analysis or for human inspection. In fact, noise contaminating an image may be a major drawback for most other image processing tasks like, for instance, image analysis, edge detection or pattern and/or object recognition and hence, it should be reduced. Similarly, increasing of resolution and image size conduces to high computational requirements, that should be decrease, especially for real time applications and so.

In the last years, the interest in using colour images has grown dramatically in a variety of applications. Therefore, colour image filtering has become an interesting area of research. It has been widely observed that colour images have to be processed taking into account the existing correlation among image channels. Probably, the most well-known approach in this sense is the vector approach. Earliest vector filtering solutions as, for instance, the vector median filter (VMF) or the vector directional filter (VDF). Unfortunately, these techniques are non-adaptive to local image statistics which implies that the processed images are usually blurred in edges and image details. To overcome this drawback, a number of adaptive vector processing solutions have been recently proposed, and a well-down approach are the *peer group* techniques. In last years, the theory of fuzzy sets (topologies, metrics and logic) has been developed, and they are a good mathematical tool to use in image filtering.

This PhD dissertation main goals are: (i) the study of fuzzy metrics applicability in colour image filtering tasks and computational improvement; (ii) the design of new colour image filtering solutions that take advantage of the observed interesting fuzzy metrics and fuzzy logic properties and obtain a good computational perform and (iii) implement parallel versions of the filters designed with MPI to test the performance with parallel computing. Extensive experimental results presented in this dissertation have shown that fuzzy metrics are useful to design filtering techniques with low computational cost which are competitive with respect to recent state-of-the-art filters.

Resumen

El filtrado de imágenes es una tarea fundamental para la mayoría de los sistemas de visión por computador cuando las imágenes se usan para análisis automático o, incluso, para inspección humana. De hecho, la presencia de ruido en una imagen puede ser un grave impedimento para las sucesivas tareas de procesamiento de imágenes como, por ejemplo, la detección de bordes o el reconocimiento de patrones u objetos y, por lo tanto, el ruido debe ser reducido. Del mismo modo, el aumento de la resolución y el tamaño de las imágenes nos conduce a requerimientos computacionales más altos, los cuales hemos de intentar rebajar sobretodo para aplicaciones en tiempo real o similares.

En los últimos años el interés por utilizar imágenes en color se ha visto incrementado de forma significativa en una gran variedad de aplicaciones. Es por esto que el filtrado de imágenes en color se ha convertido en un área de investigación interesante. Se ha observado ampliamente que las imágenes en color deben ser procesadas teniendo en cuenta la correlación existente entre los distintos canales de color de la imagen. En este sentido, la solución probablemente más conocida y estudiada es el enfoque vectorial. Las primeras soluciones que proponen técnicas de filtrado vectorial, son las conocidas técnicas del filtro de mediana vectorial (VMF) o el filtro direccional vectorial (VDF). Desafortunadamente, estas técnicas no se adaptan a las características locales de la imagen, lo que implica que habitualmente los bordes y detalles de las imágenes se emborronan y pierden calidad. A fin de solventar este problema, varios filtros vectoriales adaptativos se han propuesto recientemente, entre los que destacan las técnicas de *peer group*. En los últimos años ha aparecido la teoría de los denominados conjuntos fuzzy, borrosos o difusos (lógica, métricas y topologías), que se ha demostrado es una herramienta adecuada para el filtrado de imágenes.

En la presente Tesis Doctoral las metas principales son: (i) el estudio de la aplicabilidad de métricas difusas en tareas de procesamiento de imagen y mejora computacional; (ii) el diseño de nuevos filtros para imágenes en color que aprovechan las propiedades de las métricas fuzzy y consiguen un buen rendimiento con una gran mejora computacional y (iii) implementar versiones de los filtros diseñados con MPI para probar su rendimiento mediante computación paralela. Los resultados experimentales presentados en esta Tesis muestran que las métricas difusas son herramientas útiles para diseñar técnicas de filtrado con un bajo coste computacional y son competitivas en comparación con otras técnicas en el estado del arte.

Resum

El filtratge d'imatges és una tasca fonamental per a la majoria dels sistemes de visió per ordinador quan les imatges s'usen per a l'anàlisi automàtica o, fins i tot, per a la inspecció humana. De fet, la presència de soroll en una imatge pot ser un greu impediment per a les successives tasques de processament d'imatges com, per exemple, la detecció de vores o el reconeixement de patrons o objectes i, per tant, el soroll ha de ser reduït. De la mateixa manera, l'augment de la resolució i tamany de les imatges ens porta a requeriments computacionals més pesats, els quals hem d'intentar rebaixar especialment en aplicacions en temps reals o similars.

En els darrers anys l'interès per utilitzar imatges en color s'ha vist incrementat de forma significativa en una gran varietat d'aplicacions. És per açò, que el filtratge d'imatges en color s'ha convertit en una àrea d'investigació interessant. S'ha observat àmpliament que les imatges en color han de ser processades tenint en compte la correlació existent entre els distints canals de color de la imatge. En aquest sentit, la solució probablement més coneguda i estudiada és l'enfocament vectorial, com per exemple el filtre de mitjana vectorial (VMF) o el filtre direccional vectorial (VDF). Tot i això, aquestes tècniques no s'adapten a les característiques locals de la imatge, la qual cosa implica que usualment les vores i detalls de les imatges es difuminen i perden qualitat. A fi de resoldre aquest problema, s'han proposat recentment uns quants filtres vectorials adaptatius, entre els que destaquen les tècniques de *peer group*. Als últims anys ha aparegut la teoria dels anomenats conjunts fuzzy, borrosos o difusos (lògica, mètriques i topologies), i s'ha demostrat que és una bona eina matemàtica per al filtratge d'imatges.

En la present Tesi Doctoral els objectius han estat: (i) l'estudi de l'aplicabilitat de mètriques difuses en tasques de processament d'imatges i millora computacional; (ii) el disseny de nous filtres per a imatges en color que trauen profit de les propietats de les mètriques difuses i aconseguen un bon rendiment amb una gran millora computacional i (iii) implementar variants dels filtres dissenyats amb MPI per a provar el rendiment mitjançant computació paral·lela. Els resultats experimentals presentats en esta Tesi mostren que les mètriques difuses i la lògica difusa són eines útils per a dissenyar tècniques de filtratge amb un cost computacional baix i són competitives en comparació amb altres tècniques en l'estat de l'art.

1 Introducció

Hui en dia el consum d'imatges digitals està present a tot arreu, des del consum a nivell d'usuari com en aplicacions mèdiques, reconeixement de formes, meteorologia, enginyeria i un llarg etcètera. Les imatges, que a simple vista les veiem com una representació en dues dimensions d'un conjunt d'objectes i colors, vénen representades com *senyals* per al seu tractament digital i transmissió. La representació més comuna de les imatges és com una matriu tridimensional, de $n \times m \times x$, on n i m representen les dimensions de la imatge (alt i ample) i x és el nombre de canals o colors de la imatge. Les imatges d'un sol canal de color són les anomenades imatges en escala de grisos o en blanc i negre, mentre que les imatges en color habitualment fan servir tres canals coneguts com a RGB (Red, Green, Blue), ja que per conformar un color es realitza una combinació de Roig, Verd i Blau. Més usualment, en compte d'una matriu tridimensional, convé entendre la imatge com una matriu bidimensional, on cada element és el vector de color del que es coneix amb el nom de *píxel*.

Pel fet de ser un senyal, la imatge pot contaminar-se durant la seua adquisició o transmissió a través d'un canal. Ací és on apareix el concepte de *soroll*. Diguem que una imatge està contaminada quan alguns dels seus píxels han estat alterats o transformats, perdent la informació del color original per altres valors.

Podem considerar-ne dos tipus de soroll. Per un costat tenim el soroll que apareix en el moment de l'adquisició de la imatge per un sensor d'una càmera, escàner o similar en males condicions. Aquest soroll s'anomena *soroll blanc* o *Gaussià*. Aquest tipus de soroll té com a conseqüència l'emborronament de tots els píxels de la imatge, provocant un efecte de boira (*blur*) que difumina els píxels de la imatge. Per altra part existeix el que es coneix com *soroll impulsiu*, que apareix durant la transmissió de la imatge per un canal contaminat. Aquest tipus de soroll afecta sols a certs píxels de la imatge (a diferència del *Gaussià* que afecta a tots). L'efecte sobre els píxels contaminats és que es modifica el valor d'alguna/es de les seues components de color per valors de colors molt alts o molt baixos (com que al procés de transmissió la informació es codifica en binari, i cada component de color és un byte, l'efecte de ser modificats els valors tots a 0 representa el valor més menut, que parlant en termes de color és el negre, mentre que tots a 1 és el valor més

alt de color que és el blanc). L'efecte provocat sobre la imatge és que aquesta apareix pigallada on els píxels han estat contaminats.

Per tot l'exposat anteriorment, el filtratge d'imatges és probablement la tasca més comuna en el processament d'imatges digitals, sobretot per tal de preparar-la per altres processos, com per exemple anàlisis d'imatges, detecció de vores, reconeixement de formes i/o objectes, etc. Normalment, s'aplica el filtratge d'imatges per a eliminar o reduir el soroll que està present dins d'una imatge, i que pot alterar l'estructura de la informació que conté. De fet, el filtratge de soroll, és el procés de discriminar entre la informació original de la imatge i les dades sorolloses, la qual cosa fa que el problema del filtratge pugui tractar-se com un problema d'interpretació d'informació, és a dir discriminar la informació sorollosa de l'original. Com a conseqüència, el filtratge d'imatges es converteix en un pas fonamental dins de qualsevol sistema de visió per ordinador on les imatges es fan servir per anàlisi automàtic o inspecció humana. En aquesta Tesi Doctoral ens centrarem en l'eliminació de soroll impulsiu.

Les primeres aproximacions a les tècniques de filtratge varen ser creades per a imatges d'un sol canal o en escala de grisos, solucions que es varen fer per eliminar tipus específics de soroll. En els últims anys, l'interès en fer servir senyals multicanal, i concretament imatges en color, ha fet créixer molt la varietat d'aplicacions. Per tant, el filtratge d'imatges en color s'ha convertit en una àrea interessant d'investigació. Les primeres solucions per a filtrar imatges en color eren mètodes component a component on es feien servir les tècniques d'escala de grisos a cadascuna de les components del color individualment. Tot i això, és sabut que existeix una correlació entre els canals de la imatge que s'ha de tenir en consideració. Per altra banda, alguns artefactes de color i altres efectes col·laterals poden aparèixer a les imatges processades. Això implica la necessitat de solucions per al filtratge d'imatges específiques per a imatges en color en compte d'adaptar les tècniques d'escala de grisos.

Una de les vessants per al processament d'imatges més estudiada és la vessant vectorial. D'acord amb aquest esquema, cada píxel de la imatge es tracta com un vector format per les components de color, i la imatge es tracta com un camp de vectors. Per tant, tots els canals de la imatge es processen conjuntament, i la correlació entre els canals de la imatge necessàriament són preses en consideració. Les bases teòriques dels filtres es basen en la teoria de l'estadística *robusta*. Des d'aquest enfocament es pot afirmar que les observacions no representatives (o *outliers*) dins del conjunt de vectors es poden considerar com contaminants de la informació. Llavors aquestes tècniques de filtratge proporcionen eines per a interpretar o classificar aquests *outliers*, juntament amb mètodes per a, o bé rebutjar-los o bé aplicar procediments per reduir la seua influència. Com que els píxels *outliers* són una mostra diferenciada de la resta de píxels circumdants, aquestes mètodes es basen en la

idea de distingir-los per mitjà de distints mètodes que veurem a continuació en una breu visió històrica.

Una de les primeres solucions presentades va ser el *Vector Median Filter* (VMF [4]). En aquest filtre es creen finestres de filtratge sobre cada píxel de la imatge. Una finestra de filtratge W és una finestra quadrada de costat imparell que conté un conjunt de píxels, de forma que s'avalua el píxel central de la mateixa. Doncs per a cada píxel de la finestra W es calculen les distàncies acumulades a la resta dels píxels de la finestra. Aquestes distàncies s'ordenen a continuació de manera que aquell píxel la distància acumulada del qual a la resta és la menor es pren en consideració per a ser el millor com a píxel central de la finestra, i per tant s'efectua la substitució del píxel central pel seleccionat. Les distàncies poden ser calculades amb qualsevol mètrica ben coneguda. Aquest filtre elimina bé el soroll impulsiu, ja que els *outliers* difícilment apareixeran amb menors distàncies acumulades. El problema que presenta aquesta tècnica és l'absència d'adaptativitat, és a dir, no té en consideració si el píxel analitzat és sorollós o no, ja que filtrarà tant els píxels contaminats com els lliures de soroll. L'efecte provocat és una difuminació general de la imatge i pèrdua de detall en vores i textures fines.

Una modificació del VMF és el *Arithmetic Mean Filter* (AMF [84]). En aquest filtre simplement s'intercanvia el píxel central de la finestra de filtratge per la mitjana aritmètica de la resta de píxels de la finestra. Aquest filtre millora computacionalment el VMF, però la qualitat de la sortida és pitjor ja que no fa cap mena de selecció de quin píxel seleccionar. També manca d'adaptativitat a l'igual que el VMF, filtrant tots els píxels, i introduint els mateixos problemes.

Altra solució posterior és el filtre *Basic Vector Direccional Filter* (BVDF [113]). Seguint amb el tractament dels píxels de la imatge com a vectors, aquest filtre efectua una ordenació vectorial segons els angles entre els vectors de color. Similarment al VMF efectua el càlcul de la suma dels angles de cada píxel dins de la finestra de filtratge a tota la resta de píxels, ordena aquestes distàncies angulars acumulades i selecciona com a candidat a píxel central aquell amb menor distància angular acumulada. Presenta els mateixos inconvenients que el VMF: absència d'adaptativitat, juntament amb l'alt cost computacional.

Una combinació efectuada per tal de millorar les sortides dels filtres va aparèixer amb el *Directional Distance Filter* (DDF [47]). Aquest filtre fa servir com a criteri de distància un producte ponderat entre distància vectorial acumulada (criteri del VMF) i la distància angular vectorial (criteri del BVDF). Amb això el que aconseguim és poder discriminar vectors distants amb mateix angle (que eren considerats bons en el BVDF) així com vectors propers amb angle distant (considerats bons amb el VMF). Es millora la sortida del filtratge però no se soluciona el problema de l'adaptativitat i el cost computacional del mateix és notablement més alt ja que s'han d'efectuar molts més càlculs.

Tots els filtres exposats fins ara presenten el mateix inconvenient, l'absència d'adaptativitat. Per a superar aquest inconvenient i acoblar-se a les característiques locals de les imatges aparegueren a la literatura els filtres adaptatius. Entre ells podem destacar les tècniques basades en coeficients ponderats, on s'aplica a cada píxel dins la finestra un pes que serveix d'ajuda per computar la sortida del filtre com una mitjana ponderada dels vectors de la finestra. La complicació d'aquests filtre radica en el càlcul d'aquests pesos, ja que fan servir des d'estadístiques variades i funcions polinomials fins la intel·ligència artificial.

Altres tècniques són les anomenades tècniques de filtratge múltiple, en la que s'efectua el filtratge per distints sub-filtres alhora, de manera similar al DDF.

Unes de les tècniques adaptatives més importants hui en dia són les tècniques de filtratge per substitució, sobretot per a soroll impulsiu, que és on ens hem centrat en aquesta Tesi Doctoral. Aquestes tècniques efectuen el procés de filtratge als píxels sorollosos, mantenint així les característiques locals de les imatges (vores i vèrtexs) inalterades. La dificultat radica en la detecció dels píxels sorollosos, amb la qual cosa aquestes tècniques treballen en dues fases, una primera de detecció dels impulsos i una segona on es processen aquests píxels sorollosos. La manera més habitual de treballar és la construcció d'un *cluster* o grup d'anàlisi d'un píxel, normalment format pels píxels veïns a l'estudiat i estudiant el grau de pertinença del píxel analitzat al grup, detectant-se com a soroll quan el grau de pertinença és baix.

Com a evolució del grup d'anàlisi apareix el concepte de *peer group*. El *peer group* d'un píxel pot definir-se com el conjunt de píxels veïns propers a ell (donada una mètrica i un llindar de distància). Menció especial requereix el treball de Smolka, amb el seu *Peer Group Filter* (PGF [105]). En aquest filtre, s'estudia la cardinalitat del *peer group* de cada píxel, i si aquesta és major que un paràmetre donat es declara el central com no corrupte. Un cop classificats el píxels entre corruptes i no corruptes ja s'efectua el procés de filtratge dels sorollosos per mitjà d'alguna tècnica (habitualment AMF). Aquestes tècniques, a banda de millorar la qualitat de les sortides, com que efectuen menys càlculs són computacionalment més eficients.

Tenint en compte que dins de les imatges és complicat distingir entre soroll i píxels que representen vèrtexs, els conjunts fuzzy, capaços de treballar en incerteses o llindars, també han demostrat ser molt adequats per al filtratge d'imatges. Tanmateix, donada la demostrada eficiència computacional de les mètriques fuzzy en aquests aspectes, ens vàrem proposar el fer-les servir per la implementació dels nostres filtres. En els filtres clàssics anteriorment citats es fan servir mètriques clàssiques (habitualment la L_2 (Euclídea)). Nosaltres hem fet servir la següent mètrica fuzzy amb la que, a banda de millorar la detecció de soroll, com es veurà millora el rendiment computacional del filtratge.

Siga $(x_i(1), x_i(2), x_i(3))$ el vector de color d'un píxel \mathbf{x}_i a l'espai de color RGB, i siga X el conjunt $\{0, 1, \dots, 255\}^3$. Llavors, fixat $K > 0$, la funció $M : X \times X \rightarrow]0, 1]$ donada per

$$M(\mathbf{x}_i, \mathbf{x}_j) = \prod_{l=1}^3 \frac{\min\{x_i(l), x_j(l)\} + K}{\max\{x_i(l), x_j(l)\} + K}$$

és una mètrica fuzzy F -acotada, en el sentit de George i Veeramani, en X , que és estacionària, (Gregori i Romaguera [34]).

Aquesta mètrica compleix les propietats:

- (GV1) $M(x, y) > 0$
- (GV2) $M(x, y) = 1$ si i sols si $x = y$
- (GV3) $M(x, y) = M(y, x)$
- (GV4) $M(x, z) \geq M(x, y) \cdot M(y, z)$

Partint de tot l'exposat fins ara, i després d'un llarg procés d'experimentacions en el que buscàvem millores en el temps computacional i de qualitat en les imatges filtrades, en aquesta Tesi Doctoral es presenten tres noves tècniques de filtratge d'imatges en color de les que hem provat la seua competitivitat a nivell de qualitat en la imatge filtrada i a nivell computacional respecte als filtres clàssics, actuals i altres que li són comparables. Aquestes tècniques basades en el concepte de *peer group* estan descrites detalladament mitjançant l'algorisme emprat així com un estudi detallat del cost computacional de les mateixes. A més a més es presenten mesures objectives, visuals i mesures en quant a temps computacional de les imatges filtrades comparant-les amb les tècniques de filtratge clàssiques i recents de la literatura. Com a investigació adicional s'ha fet una implementació de les tres noves tècniques implementades mitjançant computació paral·lela, donada l'absència d'implementacions de filtres en paral·lel i per poder desenvolupar aquestes tècniques en imatges de grandària considerable on el temps de filtratge és molt significatiu.

La Tesi Doctoral es divideix en Capítols, els continguts dels quals es descriuen breument com segueix:

Capítol 2 Un estat de l'art més detallat i complet de les tècniques de filtratge d'imatges.

Capítol 3 S'entra en un estudi succint de les mètriques fuzzy així com una introducció als conjunts i topologia fuzzy. Començant pel concepte de conjunt fuzzy definit per Lofti A. Zadeh [123], en els quals s'assigna a cada objecte de l'univers del discurs un grau de pertinença entre 0 i 1 al conjunt. Aquests conjunts es combinen amb les variables lingüístiques i amb les regles de decisió (regles fuzzy *if-then*) per conformar la lògica fuzzy. Amb aquesta lògica, podem inferir a partir de valors numèrics d'entrada els graus de pertinença de les variables als conjunts, procés conegut

amb el nom de *fuzzyfication*, aplicar la inferència fuzzy per a posteriorment amb el procés invers o *defuzzyfication* tornar a valors numèrics (depenent de la naturalesa del problema). S'expliquen també al capítol les mètriques i topologies fuzzy, ja que serveixen com a base per a millores computacionals en els algorismes implementats al llarg de la Tesi Doctoral, centrant-se en les de George i Veeramani.

Capítol 4 Es presenta el filtre *Fuzzy Modified Peer Group Filter* (FMPGF). En aquest disseny el que pretenem és incrementar la velocitat dels filtres actuals considerats com a ràpids i destinats a aplicacions en temps real. Llavors prenem com a model el PGF, redefinim el concepte de *peer group* amb la mètrica fuzzy plantejada anteriorment de la següent manera. Donats un píxel central \mathbf{x}_i en una finestra W i un valor $d \in]0, 1]$ definim $\mathcal{P}(\mathbf{x}_i, d)$ com el conjunt

$$\{\mathbf{x}_j \in W : M(\mathbf{x}_i, \mathbf{x}_j) \geq d\}$$

això vol dir que $\mathcal{P}(\mathbf{x}_i, d)$ és el conjunt de píxels de la finestra de filtratge W la distància fuzzy dels quals al píxel \mathbf{x}_i no és menor que d . Amb aquest canvi ja millorem computacionalment el filtratge, però anem més enllà i ens aprofitem de l'axioma 4 de la mètrica fuzzy (desigualtat triangular per al producte) per poder diagnosticar més d'un píxel quan fem l'anàlisi d'un *peer group*. Separem la fase de diagnòstic en dos passos. En un primer pas, sobre finestres disjunctes construïm el *peer group* i en cas de tenir una cardinalitat adequada, diagnosticuem com no corrupte no sols el píxel central del *peer group*, si no tots els píxels del *peer group*. Si no s'arriba a la cardinalitat desitjada es declara el central com a corrupte. Al segon pas, sobre els píxels que no s'han pogut diagnosticar es torna a repetir el procés anterior fins diagnosticar tots els píxels. Finalment a una segona fase es filtren els píxels diagnosticats com corruptes. Amb aquest filtre aconseguim millorar el cost computacional dels filtres anteriors de la literatura, aconseguint de vegades millors resultats en la qualitat de la imatge filtrada. Com a resultat podem dir per la part del cost computacional que el nombre mitjà de distàncies per a filtrar un píxel exigint almenys 2 veïns dins del *peer group* en finestres 3×3 al nostre algorisme és de 5.844 distàncies davant de les 4.528 distàncies del filtre PGF i de les 36 distàncies del VMF. Per la part de la qualitat del filtratge el paràmetre NCD (que mesura la qualitat visual de la imatge filtrada, detallat al Capítol 2 i a menor valor millor resultat) s'obtenen resultats de 1.915, 0.739 i 0.610 (VMF, PGF i FMPGF, respectivament) per a la imatge Peppers de molta textura i poc de detall contaminada amb un 5% de soroll impulsiu. A la imatge Lenna (de detall i textures mixtes) contaminada també amb un 5% de soroll impulsiu, s'obtenen uns resultats de 1.781, 0.297 i 0.370, respectivament. Finalment en una imatge de detall alt amb poques textures com és Baboon, contaminada també amb

un 5% de soroll impulsiu, obtenim un 4.496,1.062 i 1.007. D'aquests resultats podem concloure que s'aconsegueix tant millores computacionals en totes les bateries d'experimentacions efectuades, així com millores en la qualitat de la imatge, però aquestes ja depenent de l'estructura de la imatge així com de la quantitat de soroll que aquesta conté.

Els resultats d'aquest capítol han estat publicats com a Joan-Gerard Camarena, V. Gregori, S. Morillas, A. Sapena, Fast detection and removal of impulsive noise using peer groups and fuzzy metrics, *Journal of Visual Communication and Image Representation*, 19 (2008) 20–29.

Capítol 5 Dissenyem i implementem el filtre *Fuzzy Modified Neighborhood Filter* (FMNF). L'objectiu fonamental d'aquesta tècnica és la millora computacional, per tal d'optimitzar el temps emprat en aplicacions en temps real tot i que això anirà en detriment de la qualitat de la imatge filtrada. Ara bé, aquesta pèrdua pot considerar-se acceptable en comparació amb el filtre anterior. Per aconseguir aquest guany, l'algorisme es centra en la fase de diagnòstic dels píxels, reduint el cost computacional de dita fase. A diferència del filtre anterior, la fase de diagnòstic s'efectua d'una sola passada, si bé el mecanisme de diagnòstic amb el *peer group* és el mateix que al FMPGF. Ací a la fase de diagnòstic es construeix el *peer group* sobre el primer píxel de la imatge avaluant la cardinalitat del mateix a l'igual que el mètode del capítol anterior. Posteriorment, a diferència del FMPGF on passàvem a la següent finestra de filtratge, es continua l'estudi al següent píxel de la imatge seguint l'ordre natural de la mateixa. Aquest píxel potser que ja haja estat diagnosticat a algun pas anterior (cas de pertànyer a un *peer group* adequat anteriorment), amb la qual cosa no cal fer-li un nou anàlisi i l'algorisme seguirà pel següent píxel. Aquest procés es repeteix fins completar tots els píxels de la imatge. El filtratge s'efectua sobre els píxels diagnosticats com corruptes. La millora computacional d'aquesta tècnica és considerable, mantenint una qualitat de filtratge acceptable. Alguns dels resultats que podem concloure d'aquesta tècnica són per la part del cost computacional que el nombre mitjà de distàncies per a filtrar un píxel exigint almenys 2 veïns dins del *peer group* en finestres 3×3 al nostre algorisme és de 2.94 davant de les 4.15 del filtre PGF i de les 36 del VMF. Per la part de la qualitat del filtratge el paràmetre NCD s'obtenen resultats de 1.847, 0.553 i 0.786 (VMF, PGF i FMPGF, respectivament) per a la imatge Lenna (de detall i textures mixtes) contaminada amb un 10% de soroll impulsiu. A la imatge Baboon (contaminada també amb un 10% de soroll impulsiu) s'obtenen uns resultats de 4.541, 1.439 i 1.763, respectivament. Els resultats que presentem ens diuen que la millor computacional està al voltant del 30%, sense perdre excessivament en la qualitat de la imatge filtrada, el que fa aquesta tècnica molt adequada per a processament en temps real.

Els resultats d'aquest capítol han estat publicats com a Joan-Gerard Camarena, Valentín Gregori, Samuel Morillas and Guillermo Peris-Fajarnés “New Method for Fast Detection and Removal of Impulsive Noise Using Fuzzy Metrics”, *Lecture Notes in Computer Science* vol. 1, pp. 359-369 (2006). Proceedings of the *International Conference on Image Analysis and Recognition ICIAR 2006*

Capítol 6 El filtre Improved Fast Peer Group Filter (IFPGF) és el desenvolupat en aquest capítol. L'objectiu d'aquest filtre és aconseguir una millora en la qualitat de la imatge filtrada, motiu pel qual s'efectua una fase de diagnòstic de píxels en dos passos. Al primer pas, a l'igual que el FMPGF, es construeix el *peer group* en finestres disjunctes i si la cardinalitat arriba a $m + 1$ membres (+1 tenint en compte que el central sempre és membre del *peer group*) es diagnostica el central i els membres del *peer group* com no corruptes. Si no s'arriba a la cardinalitat establerta, el central es declara corrupte. La novetat és que a la segona fase, es repeteix el procés sobre els píxels no diagnosticats, però podran diagnosticar-se com no corruptes si, o bé tenen un *peer group* de $m + 1$ elements o bé dins del *peer group* hi ha almenys m' ($m' < m$) píxels diagnosticats com no corruptes a la primera fase. Aquesta segona condició ens permet ser molt més exactes alhora del diagnòstic de píxels, cosa que es veu reflexada en els molt bons resultats en termes de qualitat de les imatges obtingudes. Entre altres resultats s'obtenen valors de NCD de 0.34 davant d'un 0.41 del PGF i d'un 2.36 del VMF en imatges amb nivells de detall normal mentre que s'obté un 1.36 davant un 1.48 del PGF i un 3.84 del VMF en imatges amb alt nivell de detall. Tot i que l'objectiu d'aquest filtre és la millora de la qualitat, el cost computacional segueix oferint uns resultats bons, millors que el FMPGF però no tan bons com el FMNF (filtre orientat exclusivament a la velocitat computacional).

Els resultats d'aquest capítol s'han enviat a la Revista *Image and Vision Computing* (editorial *Elsevier*), estant encara pendent de resolució.

Capítol 7 Fins els capítols anteriors s'han desenvolupat noves tècniques de filtratge d'imatges. Tot i la rapidesa de les tècniques de filtratge desenvolupades, quan les dimensions de les imatges són molt grans, com per exemple en imatges d'alta resolució o imatges per satèl·lit, com que el temps de filtratge depèn del tamany de la imatge, tendeixen a consumir molt de temps en el filtratge d'aquest tipus d'imatges, com es obvi. La necessitat de més potència computacional ens porta a dividir problemes grans en subproblemes més menuts mitjançant tècniques de “dividir i vèncer”, fent així implementacions per a processament paral·lel dels filtres desenvolupats. La motivació és que a la literatura apareixen modelats paral·lels d'aquest tipus de filtres, però amb escasses implementacions en el filtratge d'imatges, i això ens ha portat a efectuar aquesta tasca. La

implementació dels filtres s'ha dut a terme mitjançant les primitives de MPI (Message Passing Interface), on l'execució de diversos processos concurrents és controlada mitjançant pas de missatges (dades de control + dades de treball) Les execucions s'han dut a terme al centre de càlcul de la UPV en un *cluster* de processador, amb uns resultats molt positius. Positius no solament en que el filtre s'efectua amb bon rendiment si no que ens permeten veure on es troben els detalls del paradigma de la computació paral·lela i ens pot portar a millores futures.

Índex

Abstract	viii
Resumen	ix
Resum	x
Introducció	xi

Part I Preliminars

2 Estat de l'Art dels Filtres Vectorials Clàssics i Recents per a Imatges en Color.....	3
2.1 Tècniques de Filtratge no Lineals	4
2.2 Tècniques de Filtratge d'Ordenació Estadística	5
2.3 Esquemes d'Ordenació de Dades Multivariades	6
2.4 Tècniques Clàssiques de Filtratge Vectorial	11
2.5 Filtres d'imatges en color adaptatius	15
2.6 Avaluació Objectiva del Rendiment del Filtres d'Imatges en Color	19
3 Fonaments de Conjunts Fuzzy: Lògica, Topologia i Mètriques	25
3.1 Concepte de Conjunt Fuzzy	25
3.2 Principis de Lògica Fuzzy	26
3.3 Principis de Topologia i Mètrica Fuzzy	29
3.4 Espais Mètrics Probabilístics	30
3.5 Espais Mètrics Fuzzy de Kaleva i Seikkala	32
3.6 Espais Mètrics Fuzzy de Kramosil i Michalek	33
3.7 Espais Mètrics Fuzzy de George i Veeramani	34

Part II Algorismes Implementats

4	Detecció i Eliminació Ràpida de Soroll Impulsiu amb Peer Groups i Mètriques Fuzzy	39
4.1	Introducció	39
4.2	Una Mètrica Fuzzy Adequada	41
4.2.1	Anàlisi Computacional de la Mètrica Fuzzy	42
4.3	Detecció Proposada de Píxels Corruptes	47
4.4	Estudi Experimental i Comparativa de Rendiment	54
4.5	Complexitat Computacional	59
4.6	Conclusions	72
5	Nou Mètode per a la Detecció i Eliminació Ràpida de Soroll Impulsiu amb Mètriques Fuzzy	75
5.1	Introducció	75
5.2	Detecció i Eliminació de Píxels Corruptes	76
5.3	Estudi Experimental i Comparativa de Rendiment	79
5.4	Complexitat Computacional	84
5.5	Conclusions	92
6	Millores en el Filtratge d'Imatges Mitjançant Tècniques de Peer Group	95
6.1	Introducció	95
6.2	Detecció i Eliminació de Píxels Corruptes	97
6.3	Estudi Experimental	98
6.4	Complexitat Computacional	124
6.5	Implementació <i>fuzzy</i> del Mètode Proposat	126
6.6	Conclusions	136
7	Filtratge d'Imatges Mitjançant Computació Paral·lela amb MPI	139
7.1	Introducció	139
7.1.1	La Necessitat de Potència Computacional	140
7.1.2	La Necessitat de la Computació Paral·lela	141
7.1.3	Introducció al Filtratge d'Imatges en Color	142
7.2	Representació en Memòria d'Imatges	143
7.3	Filtratge d'Imatges	146
7.3.1	Filtres d'Una Fase	146
7.3.2	Filtres de Dos Fases	147
7.4	Paral·lelitzant els Filtres	150
7.5	Resultats Experimentals	153
7.6	Conclusions i Treball Futur	159

Part III Conclusions i Treball Futur

8	Conclusions i Treball Futur	163
----------	--	-----

Referències 165

Part I

Preliminars

2 Estat de l'Art dels Filtres Vectorials Clàssics i Recents per a Imatges en Color.

En qualsevol imatge digital a color, alguns valors del color dels píxels poden veure's modificats per la presència de soroll. L'objectiu general de les diferents tècniques de filtratge de soroll és eliminar les observacions incorrectes o, almenys, reduir la seua presència, sense modificar aquells colors que no han estat modificats pel soroll.

Habitualment es consideren dos tipus de soroll que poden afectar les imatges en color. Per una part, el soroll associat al sensor de la càmera, també anomenat soroll tèrmic, i, d'altra banda, el soroll que pot introduir-se durant la transmissió de la imatge per un canal contaminat [84]. El soroll del sensor de la càmera es modela com a soroll *Gaussià blanc*. Els errors de transmissió es modelen com a soroll *impulsiu*. El procés de contaminació del soroll impulsiu afecta sols a certs píxels de la imatge, canviant una o més components del píxel afectat per valors que són significativament desviats dels originals.

Al llarg d'anys d'investigació s'han proposat diverses tècniques de filtratge. Entre les tècniques proposades, podem trobar algunes de processament lineal que són matemàticament senzilles i poden dissenyar-se i implementar-se fàcilment. Aquestes tècniques s'han fet servir durant anys a causa de la seua simplicitat i rendiment suficient en certes aplicacions. La majoria d'aquestes tècniques treballen assumint que el senyal processat pot representar-se per mitjà d'un model estacionari i llavors, tracten d'optimitzar els paràmetres apropiats per a eixe model.

Tot i això, alguns problemes en l'àrea del processat d'imatges no poden ser tractats eficientment amb l'ús de tècniques lineals. Malauradament, les tècniques de processat lineals tenen certes carències en el tractament d'imatges, per que no poden fer front a les no linealitats en la construcció del model de la imatge i no tenen en compte les particularitats no lineals del sistema visual humà (HVS - *Human Visual System*) [77].

Les imatges estan compostes per regions planes, zones amb textures i vores marcades que ressalten la informació rellevant per a la percepció visual. Els filtres que són capaços de conservar les vores i detalls són altament adequats per al filtratge i millora d'imatges. Malauradament, la majoria de tècniques de processat de senyals tendeixen a difuminar les vores i desbaratar altres detalls de la imatge [84].

2.1 Tècniques de Filtratge no Lineals

La necessitat de tractar en sistemes complexos no lineals, conjuntament a la disponibilitat d'una major capacitat computacional, comporta la reavaluació de les tècniques convencionals de filtratge. Són necessaris nous algorismes i tècniques que aprofiten les capacitats computacionals actuals i que facen servir assumpcions més realistes. Amb eixa finalitat, han anat introduint-se les tècniques de processat de senyals no lineals. Teòricament, les tècniques no lineals són capaços d'eliminar el soroll no Gaussià, per conservar vores, detalls de la imatge i eliminar defectes introduïts durant la creació (mitjançant escàner o càmeres) o transmissió a través de canals no lineals. A pesar del creixement recent d'aquest tipus de tècniques i l'aparició de nous resultats teòrics, eines i aplicacions, les tècniques de filtratge no lineal no disposen encara d'una teoria unificadora. Per contra, cada tipus d'operadors no lineals tenen les seues eines matemàtiques pròpies que proporcionen una anàlisi de rendiment raonablement bo. Com a resultat, han aparegut publicades varies tècniques no lineals de processat de senyals. Hui dia les següents classes de tècniques de processat no lineals poden identificar-se [84]:

- Tècniques basades en polinomis
- Tècniques homomòrfiques
- Tècniques basades en morfologia matemàtica
- Tècniques basades en l'ordre estadístic

Els filtres polinomials, especialment els filtres de segon ordre de Volterra (quadràtics), s'han fet servir per filtratge d'imatges en color, modelat de canals no lineals en telecomunicacions i per a processat de senyals multicanal.

Els filtres homomòrfics i les seues variacions són unes de les primeres classes de filtres no lineals i s'han fet servir en imatges digitals i processament de senyals. Aquest tipus de filtre s'ha utilitzat en aplicacions com per exemple eliminació de soroll depenent multiplicatiu, processament d'imatges a color i processament d'imatges de satèl·lit multicanal. La seua característica bàsica és que usen no linealitats, principalment logaritmes, per a transformar les no linealitats associades al senyal en senyals acumulatius i llavors el seu procés amb filtres lineals. La sortida del filtre lineal pot transformar-se mitjançant l'operació inversa no lineal.

La morfologia matemàtica pot descriure's geomètricament segons les accions preses o els operadors aplicats sobre imatges binàries, monocrom o a color. La descripció geomètrica depèn d'imatges sintètiques menudes anomenades elements estructurals. Aquesta varietat de morfologia matemàtica, de vegades anomenada morfologia estructural, és adequada per a l'anàlisi i procés d'imatges. Podem trobar els filtres morfològics en processament d'imatges i aplicacions d'anàlisi. Específicament, les seues àrees d'aplicació inclouen filtratge d'imatges, millora d'imatges i detecció de vores.

A la secció següent, es descriu la família de filtres d'ordenació estadística. Aquesta família de filtres és probablement la família més popular de filtres no lineals i és la família de filtres involucrats al treball fet en aquesta dissertació.

2.2 Tècniques de Filtratge d'Ordenació Estadística

Les bases teòriques dels filtres d'ordenació estadístics és la teoria de l'estadística robusta [20]. Existeixen gran varietat de filtres per a imatges en color dins d'aquesta família, sent el *Filtre de Mediana Vectorial* (VMF) el més conegut [4]. El concepte de rerefons d'aquest enfocament és que les observacions no representatives o *outliers* en conjunts de vectors de color poden considerar-se com contaminats de la informació i llavors poden presentar un inconvenient per al procés en tasques posteriors. Per tant, els filtres d'ordre estadístic proporcionen eines per a interpretar o classificar els *outliers* (píxels que destaquen dels seus veïns, normalment considerats com a soroll) i mètodes per tractar-los, be rebutjant-los o aplicant procediments per reduir la seua influència. Els *outliers* poden trobar-se en mostres de dades escalars (univariant) encara que també existeixen en dades multivariants com vectors de colors.

El concepte bàsic de *outlier* és una observació estadísticament inesperada en termes d'algun model bàsic que pot estendre's a dades multivariants i, en particular, a imatges en color. No obstant, l'expressió d'aquesta noció i la determinació del procediment adequat per identificar un *outlier* no és directa quan l'operació es realitza a dades multivariants, principalment pel fet que un *outlier* en dades multivariants no té una manifestació simple com una observació que es desvia un màxim respecte la resta de mostres.

En anàlisi de dades univariants o univariables, existeix un ordre natural de les dades que fa possible identificar valors extrems i la distància d'aquests extrems al centre pot calcular-se fàcilment. D'aquesta forma, el problema d'identificar i aïllar valors solts que són atípics davant la resta del conjunt és simple. Per aquest motiu, han aparegut a la literatura nombroses tècniques de filtratge basades en l'ordre de dades univariants.

La popularitat i l'extens ús de filtres d'ordre estadístics escalars ens porten a la introducció de tècniques similars per a l'anàlisi de dades multivariables i senyals multicanal, com vectors de colors i imatges en color. No obstant, en tal de dissenyar aquesta mena de filtres, hem de solucionar el problema d'ordenar dades multivariables.

Als següents apartats, presentem algunes tècniques bàsiques de resoldre el problema d'ordenar dades multivariables i alguns filtres vectorials clàssics que fan servir les tècniques esmentades.

2.3 Esquemes d'Ordenació de Dades Multivariades

Un senyal multivarià és un senyal on cada mostra té múltiples components. Aquesta classe de senyal també s'anomena vector-valor, multicanal o multiespectral. Les imatges en color són exemples típics de senyals multicanal. Una imatge en color digital és normalment representada per les tres primàries a l'espai de color RGB com un senyal bidimensional trivaluat (tres canals). Una imatge en color a l'espai RGB normalment es representa com una matriu tridimensional $N_1 \times N_2 \times 3$, on cada píxel \mathbf{x}_i ($i \in N_1 \times N_2$) és un vector de tres components de valors enters prenent valors a l'interval $[0, 255]$, representant respectivament les components Roja, Verda i Blava (Red-Green-Blue). Les components $N_1 \times N_2$ representen les dimensions de la imatge.

Siga \mathbf{X} una variable aleatòria p -dimensional i $\mathbf{X} = [X_1, X_2, \dots, X_p]^\tau$ un vector p -dimensional de variables aleatòries i siguin $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ n mostres aleatòries en \mathbf{X} . Cada mostra \mathbf{x}_i és un vector p -dimensional d'observacions $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{ip}]^\tau$. L'objectiu és ordenar els n valors $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ d'acord a algun model d'ordenació.

El concepte d'ordre de dades natural en dades univariades no pot estendre's directament al cas multidimensional ja que no existeix una forma no ambigua, acceptada universalment, d'ordenar n mostres multivariades. No obstant això, hi ha maneres d'ordenar dades multivariades, que són els *principis de subordre* [6, 7], que es classifiquen en quatre tipus:

- Ordre Marginal o M-ordre
- Ordre Condicional o C-ordre
- Ordre Parcial o P-ordre
- Ordre Reduït (agregat) o R-ordre

M-ordre

Segons l'esquema d'ordre marginal o M-ordre, les mostres s'ordenen en cadascuna de les p -dimensions independents, obtenint:

$$\begin{aligned} \mathbf{x}_{1(1)} &\leq \mathbf{x}_{1(2)} \leq \dots \leq \mathbf{x}_{1(n)} \\ \mathbf{x}_{2(1)} &\leq \mathbf{x}_{2(2)} \leq \dots \leq \mathbf{x}_{2(n)} \\ &\dots \\ \mathbf{x}_{p(1)} &\leq \mathbf{x}_{p(2)} \leq \dots \leq \mathbf{x}_{p(n)} \end{aligned}$$

on $\mathbf{x}_{i(k)}$ representa el valor del i -èssim canal posicionat a la k -èssima posició.

Amb tot això, el vector $\mathbf{x}_1 = [x_1(1), x_2(1), \dots, x_p(1)]^\tau$ està format pels elements mínims de cada dimensió i el vector $\mathbf{x}_n = [x_1(n), x_2(n), \dots, x_p(n)]^\tau$ està format pels elements màxims de cada dimensió, on τ representa l'operació transposta de la matriu fila. La mitjana marginal es defineix com

$$\mathbf{x}_{\nu+1} = [x_1(\nu + 1), x_2(\nu + 1), \dots, x_p(\nu + 1)]^\tau$$

per a $n = 2\nu + 1$. Noteu que aquest vector pot no correspondre's amb cap de les dades originals. En canvi, al model escalar, hi ha una relació un a un entre les mostres originals \mathbf{x}_i i l'ordre estadístic $\mathbf{x}(i)$.

C-ordre

Al cas de l'ordre condicional o C-ordre, les mostres de dades multivariades s'ordenen condicionalment en un dels conjunts d'observacions marginals. Així, una de les components marginals s'ordena, i les altres components de cada vector s'ordenen en funció de la posició de la component ordenada prèviament. Suposant que la primera dimensió és l'ordenada, la resta de les mostres quedarien:

$$\begin{aligned} \mathbf{x}_{1(1)} &\leq \mathbf{x}_{1(2)} \leq \dots \leq \mathbf{x}_{1(n)} \\ \mathbf{x}_{2[1]} &\leq \mathbf{x}_{2[2]} \leq \dots \leq \mathbf{x}_{2[n]} \\ &\dots \\ \mathbf{x}_{p[1]} &\leq \mathbf{x}_{p[2]} \leq \dots \leq \mathbf{x}_{p[n]} \end{aligned}$$

on $\mathbf{x}_1(i), i = 1, 2, \dots, n$ són els ordres estadístics marginals de la primera dimensió, i $\mathbf{x}_j[i], j = 2, \dots, p, i = 1, 2, \dots, n$ són les mostres quasi-ordenades de les dimensions $j = 2, \dots, p$ condicionades a l'ordre marginal de la primera dimensió. De fet, aquestes components no estan ordenades, sinó, simplement llistades d'acord a la seua component ordenat en cada cas. En el cas bidimensional, la segona dimensió, no ordenada, s'anomena *concomitant* respecte a la primera dimensió, ordenada.

El principal avantatge del C-ordre és la simplicitat, ja que sols necessitem computar un ordre escalar. El desavantatge és que, ja que sols es fa servir la informació d'un dels canals per a l'ordenació, se suposa que la majoria de la informació està associada al canal ordenat. Si aquesta assumpció no s'acompleix llavors es perdrà informació important. Com a exemple, podem considerar el problema d'ordenar senyals a l'espai de color YIQ. Si es realitza un C-ordre al canal de luminiscència (Y) llavors la informació cromàtica als canals I i Q s'ignoren a l'ordenar. Per tant, qualsevol avantatge en la identificació de *outliers* a partir de la informació cromàtica es perdria.

P-ordre

D'acord al model del P-ordre, els subconjunts de dades s'agrupen formant *embolcalls convexes mínimes*. El primer embolcall convexa es forma tal que el perímetre conté un número mínim de punts i l'embolcall resultant conté tots els altres punts pertanyents al conjunt. Els punts al llarg d'eixe perímetre

s'anomenen com al *grup de p-ordre 1*. Aquests punts constitueixen el grup més extrem. Els punts del perímetre es descarten i el procés es repeteix de nou. Els punts del nou perímetre s'anomenen *grup de p-ordre 2* i s'eliminen de nou per tal de seguir el procés repetidament.

El procediment del P-ordre pot fer-se servir per aïllar *outliers* però aquest ordre no proporciona cap ordre entre els grups, i per tant no és fàcilment expressable en termes analítics. A més a més, la determinació del *convex hull* és difícil, tant conceptualment com computacionalment, especialment quan treballem amb dades amb una dimensió elevada. Per tot això, el P-ordre es considera inadequat per a implementar-lo en processat d'imatges en color.

R-ordre

Al R-ordre o ordre reduït (agregat), cada mostra multivariada \mathbf{x}_i es redueix a un valor escalar per mitjà de certa combinació dels valors de les components de la mostra. Llavors pot aplicar-se un ordre univariat als valors escalars resultants. Així, el conjunt $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ pot ordenar-se d'acord als valors escalars associats $R_i = R(\mathbf{x}_i), i = 1, 2, \dots, n$.

En contra del M-ordre, l'objectiu del qual era ordenar les mostres en cadascuna de les p -dimensions, l'objectiu del R-ordre és efectuar alguna mena d'ordre global sobre les dades multivariades originals. Per mitjà d'aquest ordre, l'ordre de dades multivariades es redueix a una operació simple d'ordenació sobre un conjunt fruit d'una transformació. Aquest ordre no pot interpretar-se com a l'ordre clàssic entre dades escalars perquè no existeix una mostra que siga mínim o màxim d'un conjunt de dades. Atès que l'ordre multivariat es basa en l'ús d'una funció de reducció $R(\cdot)$, els punts que divergeixen del *centre* en direccions oposades poden estar en la mateixa posició. A més a més, al fer servir una funció de reducció per a fer l'ordre multivariat, pot perdre's informació important. Donat el fet que les funcions de distàncies tenen un mecanisme natural per identificar els *outliers*, la funció de reducció més utilitzada és la distància generalitzada de Mahalanobis [7]:

$$R(x, \bar{x}, \Gamma) = (x - \bar{x})^T \Gamma^{-1} (x - \bar{x})$$

on \bar{x} és un paràmetre d'ubicació del conjunt de dades, o distribució subjacent, en consideració i la matriu Γ és un paràmetre de dispersió, fent servir Γ^{-1} com un coeficient de ponderació dels components de les mostres multivariades relacionat inversament la variabilitat de la població. Els paràmetres de la funció de reducció poden prendre valors arbitraris, com $\bar{x} = 0$ i $\Gamma = 1$ o poden assignar-se a la vertadera mitjana μ o dispersió σ . Si aquests valors són desconeguts, poden fer-se servir en el seu lloc les seues estimacions estàndards:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$s = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^\tau$$

Deponent del paràmetre d'ubicació utilitzat al procés d'ordenació, podem distingir entre els següents models:

a) R-ordre de mitjana:

Donat un conjunt de mostres multivariades $\mathbf{x}_i, i = 1, 2, \dots, n$ dins d'una finestra de processament i sent $\bar{\mathbf{x}}$ la mitjana de les mostres de la finestra, definim el R-ordre de mitjana com:

$$(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(n)} : \bar{\mathbf{x}})$$

on $(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(n)})$ és l'ordre definit per

$$d_i^2 = (\mathbf{x}_i - \bar{\mathbf{x}})^\tau (\mathbf{x}_i - \bar{\mathbf{x}})$$

i $(d_{(1)}^2 \leq d_{(2)}^2 \leq \dots \leq d_{(n)}^2)$.

b) R-ordre de mitjana marginal:

Donat un conjunt de n dades multivariades $\mathbf{x}_i, i = 1, 2, \dots, n$ dins d'una finestra de processament i sent \mathbf{x}_m la mitjana marginal del conjunt considerat, el R-ordre de mitjana marginal es defineix com:

$$(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(n)} : \mathbf{x}_m)$$

on $(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(n)})$ és l'ordre definit per

$$d_i^2 = (\mathbf{x}_i - \mathbf{x}_m)^\tau (\mathbf{x}_i - \mathbf{x}_m)$$

i $(d_{(1)}^2 \leq d_{(2)}^2 \leq \dots \leq d_{(n)}^2)$.

c) R-ordre de mostra central:

Donat un conjunt de n mostres multivariades $\mathbf{x}_i, i = 1, 2, \dots, n$ dins d'una finestra de processament i sent \mathbf{x}_c la mostra central de la finestra, el R-ordre de mostra central queda definit com:

$$(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(n)} : \mathbf{x}_c)$$

on $(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \dots, \mathbf{x}_{(n)})$ és l'ordre definit per

$$d_i^2 = (\mathbf{x}_i - \mathbf{x}_c)^\tau (\mathbf{x}_i - \mathbf{x}_c)$$

i $(d_{(1)}^2 \leq d_{(2)}^2 \leq \dots \leq d_{(n)}^2)$.

El model de R-ordre és especialment adequat per a la detecció de *outliers* en mostres de dades multivariades. A més a més, a diferència del M-ordre, les dades es tracten com a vectors, en compte de processar cada component separatament. A diferència del C-ordre, el model del R-ordre dóna la mateixa importància a cada canal de color. Finalment, el R-ordre és més simple que el P-ordre i més fàcil d'implementar. En conclusió, el R-ordre és el principi de sub-ordre més utilitzat dins l'anàlisi de dades multivariades i, particularment, dins del processament d'imatges multicanal.

Un procediment adequat d'ordenació de vectors per al processament d'imatges en color

Els principis de sub-ordre explicats anteriorment poden fer-se servir per a classificar qualsevol tipus de dades multivariades. Tot i això, per a definir un esquema d'ordre que resulte apropiat per al processament d'imatges en color, el procés hauria de centrar-se a ordenar vectors d'imatges en color. Per tant el criteri d'ordenació ha de satisfer els següents criteris [84]:

1. L'esquema ha de ser útil des d'una perspectiva d'estimació robusta, permetent l'extensió de les operacions d'ordre estadístics escalars al domini multivariat.
2. L'esquema ha de mantenir el concepte de nivells variables de *llunyania* present al cas d'ordre escalar.
3. L'ordenació proposada ha de tenir en compte el tipus de dades multivariades a tractar. Així, com que l'espai de color RGB normalment serà el més utilitzat, es donarà la mateixa rellevància als tres canals, i tota la informació present en cada canal serà considerada.

Basant-se en aquests tres criteris, l'esquema d'ordre que s'ha proposat i extensament s'ha fet servir a la literatura és una variació del R-ordre que fa servir certes mesures de distància adequades o similituds entre vectors de colors [4, 47, 63, 71, 84, 114]. Per a cada mostra dins del conjunt \mathbf{x}_i es defineix una mesura agregada respecte a totes les altres mostres del conjunt, definida com segueix:

$$R_a(\mathbf{x}_i) = \sum_{j=1}^n R(\mathbf{x}_i, \mathbf{x}_j)$$

Aquesta mesura agregada $R_{ai} = R_a(\mathbf{x}_i)$ es fa servir per definir l'ordre dels vectors, tal que:

$$\mathbf{x}_{(1)} \leq \mathbf{x}_{(2)} \leq \dots \leq \mathbf{x}_{(n)}$$

si i sols si

$$R_{a1} \leq R_{a2} \leq \dots \leq R_{an}$$

Fent servir esquema d'ordre, les mostres ordenades $\mathbf{x}_{(i)}$ tenen una relació un a un respecte a les mostres originals. Això és una diferència important

respecte del M-ordre. A més a més, totes les components de les mostres tenen la mateixa importància, a diferència del C-ordre.

L'esquema d'ordre proposat se centra a tenir en compte les relacions entre les mostres multivariades ja que calcula distàncies o similituds entre parelles de mostres en conjunts de dades. La sortida del procés d'ordre depèn críticament tant del conjunt de dades com de la funció $R(\mathbf{x}_i, \mathbf{x}_j)$ utilitzada per a calcular les distàncies o similituds entre parelles de mostres.

2.4 Tècniques Clàssiques de Filtratge Vectorial

Les tècniques clàssiques de filtratge vectorial es basen en els esquemes d'ordre anteriorment vistos per ordenar les mostres de dades fent servir distintes funcions de distància o similitud. Aquests mètodes es descriuen breument a continuació.

Filtre de Mediana Vectorial VMF [4]

Siga I una imatge multicanal i W la finestra de procés de tamany finit n , és a dir conté n píxels. Els vectors de la imatge dins la finestra W s'expressen com $I_j, j = 1, \dots, n$. La *distància* entre dos vectors I_i, I_j es denota com $\rho(I_i, I_j)$ on ρ és normalment una mètrica clàssica. Per a cada vector en la finestra, es calcula la distància acumulada respecte a tots els altres vectors de la finestra. El valor escalar

$$R_i = \sum_{j=1}^n \rho(I_i, I_j)$$

és la distància associada al vector I_i . Calculats es valors R_i , aquests són ordenats de manera creixent, anomenats ara $R_{(j)}$, per tal d'ordenar els vectors I_i . Així l'ordre de les distàncies R_i :

$$R_{(1)} \leq R_{(2)} \leq \dots \leq R_{(n)}$$

on $R_{(k)}$ representa el valor classificat a la k -èssima posició, determina l'ordre dels vectors I_i :

$$I_{(1)} \leq I_{(2)} \leq \dots \leq I_{(n)}$$

mitjançant l'aplicació $R_i \rightarrow R_{(j)}$ si i sols si $I_i \rightarrow I_{(j)}$

Donat aquest ordre, la sortida del VMF és $I_{VMF} = I_{(1)}$.

Al disseny del VMF, la distància entre dos vectors de color es calcula normalment fent servir alguna mesura de distància derivada de la mètrica generalitzada de Minkowski o norma L_γ que es defineix com:

$$\rho(I_i, I_j) = \left(\sum_{k=1}^m |I_i(k) - I_j(k)|^\gamma \right)^{\frac{1}{\gamma}}$$

on γ representa la mètrica utilitzada. La mètrica de Minkowski inclou la mètrica *city-block* ($\gamma = 1$) o normal L_1 , la mètrica Euclídea ($\gamma = 2$) o L_2 i la mètrica *chess-board* ($\gamma = \infty$) o norma L_∞ . Altres mesures de distància o similitud es revisen a [63, 84].

Atès que la sortida del VMF és la mostra associada a la mínima distància acumulada, VMF minimitza la distància entre les altres mostres en la finestra de filtratge. VMF pot derivar-se bé com una estimació de màxima probabilitat quan les densitats de probabilitat subjacent de les mostres d'entrada són bi-exponencials o bé mitjançant l'ús de tècniques d'ordenació estadístiques vectorials.

Per tant, el VMF és invariant en quant a escala, translació i rotació [84]. De fet, si la dimensió del vector és 1 llavors VMF se simplifica a la mediana escalar. Atès que l'impuls de la resposta del VMF és zero, elimina excel·lentment el soroll impulsiu [4, 63]. Altres aproximacions s'han fet en tal de agilitzar la velocitat del VMF mitjançant una aproximació lineal de la distància Euclídea [8] i dissenyant un algorisme ràpid quan es fa servir la norma L_1 [9]. Per altra part, el VMF ha estat estès amb números fuzzy en [17], mitjançant certes mètriques fuzzy.

Filtre de Mediana Vectorial Estès EVMF [4, 117]

La combinació del VMF amb tècniques lineals s'ha fet servir per millorar el rendiment en l'eliminació de soroll Gaussià [4, 117]. El filtre es construeix com a una combinació del VMF i del filtre de mitjana aritmètica (AMF) rebent el nom de filtre de mitjana vectorial estès (EVMF) [4, 84, 117]. Aquest filtre selecciona entre la sortida del VMF I_{VMF} i la sortida del AMF \bar{T} d'acord a la regla:

$$I_{EVMF} = \left\{ \begin{array}{ll} \bar{T} & \text{if } \sum_{j=1}^n \rho(\bar{T}, I_j) \leq \sum_j 1^n \rho(I_{VMF}, I_j) \\ I_{VMF} & \text{en altre cas} \end{array} \right\}$$

on \bar{T} indica la sortida del AMF calculada com $\bar{T} = \frac{1}{n} \sum_{j=1}^n I_j$. Al igual que el cas del VMF, poden fer-se servir diferents mètriques entre els vectors de color. El EVMF es comporta com el VMF prop de vèrtexs i zones amb alt nivell de detall. Amb tot això, evita la difuminació que podria generar-se pel AMF en aquests casos. A les parts homogènies tendeix a triar la mitjana del conjunt de vectors com a sortida del filtre, millorant la supressió de soroll.

Filtre Bàsic Direccional Vectorial BVDF [114]

El filtratge direccional fa servir una tècnica d'ordenació vectorial que mesura l'angle entre dos vectors de color com a criteri de distància per a l'ordenació. Ja que les mostres multicanals són vectors, es caracteritzen per la seua magnitud i direcció. Aquestes característiques poden fer-se servir per a dissenyar filtres d'imatges.

Al BVDF s'associa a cada vector una mesura de distància angular acumulada

$$\alpha_i = \sum_{j=1}^N A(\mathbf{I}_i, \mathbf{I}_j) \quad i = 1, 2, \dots, N,$$

on

$$A(\mathbf{I}_i, \mathbf{I}_j) = \arccos\left(\frac{\mathbf{I}_i \cdot \mathbf{I}_j}{|\mathbf{I}_i| \cdot |\mathbf{I}_j|}\right)$$

representa l'angle entre els vectors \mathbf{I}_i i \mathbf{I}_j .

Fent servir α_i com a criteri d'ordre:

$$\alpha_{(1)} \leq \alpha_{(2)} \leq \dots \leq \alpha_{(N)},$$

l'ordre de les mostres en la finestra és

$$I_{(1)} \leq I_{(2)} \leq \dots \leq I_{(N)}$$

El menor ordre estadístic associat a la menor distància angular acumulada representa la sortida del BVDF $I_{BVDF} = I_{(1)}$.

Ja que la sortida del BVDF és la mostra del conjunt més propera a la resta de vectors en termes de distància angular, i com que la direccionalitat està associada a la cromaticitat a l'espai RGB, el BVDF millora al VMF en termes de conservació de la cromaticitat.

Filtre Vectorial Direccional Generalitzat GVDF

A l'esquema del filtre vectorial direccional generalitzat GVDF, s'ordenen les r primeres mostres, i es defineix com segueix:

$$I_{GVDF} = \{I_{(1)}, I_{(2)}, \dots, I_{(r)}\}$$

La sortida del GVDF és un conjunt de r mostres l'angle de les quals α_i , $i = 1, 2, \dots, N$ és relativament menut a aquells vectors amb direccions atípiques, i per tant cromaticitats atípiques, els quals són eliminats del conjunt. Per tal de escollir la sortida final del filtre, GVDF s'utilitza conjuntament amb algun altre filtre de procés vectorial d'acord a la seua magnitud. Habitualment, GVDF es fa servir com a filtre de segon ordre ja que la seua sortida és una entrada per algun altre filtre que calcularà la sortida final.

Filtre Direccional de Distància DDF [47]

La tècnica del filtre direccional de distància fa servir simultàniament els criteris del VMF i del BVDF alhora. Això implica que aquest filtre és computacionalment més costós que el VMF i el BVDF. Per altra part, com que fa servir tant la magnitud com l'angle del vector, el DDF millora el VMF i el BVDF ja que és capaç de rebutjar vectors tant en magnituds com en direccions atípiques del conjunt de mostres i llavors, realitza una sortida més robusta.

El criteri de distància fet servir pel DDF s'expressa com un producte ponderat de la distància acumulada de Minkowski i la distancia angular acumulada com s'expressa:

$$\Omega_i = \left(\sum_{j=1}^N \left(\sum_{k=1}^m |I_i(k) - I_j(k)|^\gamma \right)^{\frac{1}{\gamma}} \right)^{1-p} \cdot \left(\sum_{j=1}^n A(\mathbf{I}_i, \mathbf{I}_j) \right)^p$$

per a $i = 1, 2, \dots, n$ i $p \in [0, 1]$. La sortida del DDF és la mostra $I_{(1)}$ associada al $\Omega_{(1)}$ mínim tal que

$$\Omega_{(1)} \leq \Omega_{(2)} \leq \dots \Omega_{(N)}.$$

A l'expressió anterior, p és un paràmetre que ajusta el criteri de la magnitud davant la direcció. Si $p = 0$, el DDF es comporta com el VMF mentre que per a $p = 1$ el DDF es comporta com el BVDF.

Filtres Híbrids

La introducció del filtre DDF estava basada en un conjunt de filtres heurístics que intenten minimitzar les funcions de distància fetes servir pel VMF i pel BVDF simultàniament. Els filtres híbrids direccional HVFs [84] treballen amb la direcció i la magnitud dels vectors de forma independent, i després apliquen una combinació d'aquests per a generar la sortida.

El filtre HVF_1 realitza una combinació lineal del VMF i el BVDF d'acord a la següent regla:

$$I_{HVF_1} = \left\{ \begin{array}{l} I_{VMF} \quad \text{si } I_{VMF} = I_{BVDF} \\ \left(\frac{|I_{VMF}|}{|I_{BVDF}|} \right) I_{BVDF} \quad \text{en altre cas} \end{array} \right\}$$

on I_{VMF} representa la sortida del VMF, I_{BVDF} representa la sortida del BVDF i $|\cdot|$ representa el mòdul del vector.

Altre filtre híbrid més complex és el HVF_2 [84] que a més a més fa servir el AMF com segueix:

$$I_{HVF_2} = \left\{ \begin{array}{ll} I_{VMF} & \text{si } I_{VMF} = I_{BVDF} \\ I_{out1} & \text{si } \sum_{i=1}^N |\mathbf{x}_i - I_{out1}| \leq \sum_{i=1}^N |\mathbf{x}_i - I_{out2}| \\ I_{out2} & \text{en altre cas} \end{array} \right\}$$

on

$$I_{out1} = \left(\frac{|I_{VMF}|}{|I_{BVDF}|} \right) I_{BVDF},$$

$$I_{out2} = \left(\frac{|I_{AMF}|}{|I_{BVDF}|} \right) I_{BVDF}$$

i I_{AMF} és la sortida del AMF.

El filtre híbrid HVF_1 i HVF_2 són capaços de millorar tant el VMF com el BVDF ja que generen un vector de sortida adequat tant des del punt de vista de la magnitud com des del punt de vista de la direcció. Per contra, tenen molt més cost computacional, ja que necessiten calcular les sortides del VMF i del BVDF en el cas del HVF_1 i del AMF i del BVDF en el cas del HVF_2 .

2.5 Filtres d'imatges en color adaptatius

El filtre vectorial clàssic anomenat a la secció anterior tenen l'inconvenient que les operacions fetes a qualsevol posició de la imatge són fixes, és a dir, no s'adapten a les característiques locals. S'ha observat que els processos no adaptatius produeixen com a resultat no desitjat la difuminació de vèrtexs i detalls propis de la imatge. Per a superar aquest inconvenient, s'han proposat diverses tècniques vectorials per adaptar el filtre a la variació de les característiques de la imatge i estadístiques de soroll, així com per obtenir un bon rendiment en aplicacions de la vida real, com per exemple procés de *microarrays*, tractament d'imatges de televisió, restauració virtual d'obres d'art i procés de vídeo en color. Alguns treballs recents d'esquemes de procés vectorials poden trobar-se a [63, 71, 72, 84]. Dins d'aquesta secció resumirem algunes de les tècniques de filtre adaptatius d'imatges en color més conegudes i/o recents que estan relacionades estretament amb aquesta Tesi Doctoral.

El procés adaptatiu d'imatges en color s'ha desenvolupat fent servir distintes tècniques. Les implementacions més recents poden classificar-se d'acord a la tècnica emprada en buscar l'adaptativitat. Entre aquestes tècniques podem trobar:

- a) Tècniques basades en coeficients ponderats
- b) Tècniques que fan múltiples filtrats
- c) Tècniques de millora per a soroll Gaussià

- d) Tècniques de filtratge per substitució
- e) Tècniques de filtratge fuzzy.

A les seccions següents comentarem en quina situació estan certs treballs pertanyents a aquestes classes.

Tècniques Basades en Coeficients Ponderats

Certes tècniques adaptatives recents es basen en el càlcul de pesos que s'assignen a cada píxel dins la finestra de filtratge. Després, els pesos calculats es fan servir per computar la sortida de forma que el soroll es redueix i les estructures bàsiques del senyal, vèrtexs i detalls subtils, es preserven. La sortida habitualment es calcula com una mitjana ponderada dels vectors de la finestra o com un procediment de mitjana vectorial ponderada.

La principal diferència entre els mètodes basats en coeficients ponderats és la tècnica emprada en calcular les ponderacions. La tècnica aplicada en [56] fa servir un algorisme d'optimització *off-line* per calcular els pesos adequats. Els treballs en [61, 66, 67, 110] fan servir estadístiques variades per a calcular els pesos. A [5] trobem l'ús de funcions polinomials. El treball de [60] calcula els pesos per a obtenir una millor conservació de la cromaticitat i el mètode de [70] calcula els pesos mitjançant computació evolucionària (vessant de la intel·ligència artificial amb processos iteratius que simula el creixement biològic).

Les tècniques de filtratge basades en ponderació de coeficients són adequades quan les estadístiques del soroll i de la imatge són desconegudes ja que aquestes tècniques poden adaptar-se per a eliminar diversos tipus de soroll.

Tècniques de Filtratge Múltiple

Certes tècniques de filtratge col·laboren calculant la sortida mitjançant distints sub-filtres alhora o dividint el procés en dos o més passos. En lloc de intentar trobar un mètode capaç de proporcionar una sortida adequada en qualsevol circumstància, aquestes tècniques fan servir diferents mètodes de sub-filtratge o fases de filtratge i amb les sortides individuals de cadascun es calcula la sortida final.

Dins d'aquestes tècniques, els mètodes de [75, 52, 53, 116] proposen fer servir simultàniament distints filtres en cada localització de la imatge. Llavors, en [75] la sortida del filtre es calcula prenent la millor sortida d'entre les sortides de cada sub-filtre, i en [52, 53, 116] mitjançant una fusió de les sortides dels sub-filtres. Als treballs de [52, 53] es fa servir una funció racional per elaborar la fusió i al treball de [116] s'utilitza un algorisme genètic. El mètode de [73] busca similituds entre els vectors de color del veïnat en un procediment en dos fases de detecció d'impulsos i, s'ha introduït recentment en [74] un procediment de tres fases incloent estimacions robustes, classificació de vectors i filtratge ponderat.

Ja que s'obté un rendiment flexible amb les distintes tècniques de sub-filtratge o filtratge en passos, aquest tipus de filtres poden fer-se servir quan les característiques de la imatge o del soroll siguin desconegudes.

Tècniques de Suavitat de Soroll Gaussià

L'anomenat soroll Gaussià és un altre tipus de soroll que normalment corromp les imatges digitals. El soroll Gaussià additiu, que normalment s'introdueix al procés d'adquisició de la imatge, es caracteritza per afegir a cada canal dels píxels de la imatge un valor aleatori d'una distribució gaussiana de mitjana zero. La variància d'aquesta distribució determina la intensitat i grau de corrupció del soroll. La propietat de mitjana zero permet eliminar aquest soroll calculant valors mitjans dels canals del píxel.

Idealment, eliminar soroll Gaussià implica suavitzar les diferents àrees de la imatge sense degradar ni les formes dels vèrtexs ni els detalls. Els filtres lineals clàssics, com el filtre de mitjana aritmètica (AMF) o el filtratge Gaussià, suavitzen el soroll però difuminen bastant els vèrtexs. S'han fet servir mètodes no lineals adaptatius en tal de solucionar el problema. L'objectiu dels mètodes proposats en la literatura és detectar vèrtexs mitjançant mesures locals i suavitzar-los menys que la resta de la imatge, per a conservar millor la seua forma. Un mètode ben conegut és la difusió anisotròpica introduïda en [82]. En aquesta tècnica, la variació de la imatge local es calcula en cada punt i píxels del veïnat, el tamany i forma dels quals depenen de la variació local calculada. Aquests mètodes de difusió són intrínsecament iteratius, ja que fan servir equacions diferencials per al càlcul. Un mètode no iteratiu interessant, és el filtre bilateral (BF) estudiat a [112]. La sortida del BF a una localització concreta és la mitjana ponderada dels píxels del veïnat, on els pesos per a cada píxel depenen de la proximitat espacial o similitud fotomètrica respecte al píxel en procés. El BF s'ha fet servir tractant adequadament l'eliminació de soroll Gaussià, i ha estat objecte d'estudis posteriors [24, 29, 97]. Als treballs [24, 29, 97] es proposen altres tècniques per calcular els pesos utilitzats en la ponderació.

Les tècniques mencionades anteriorment estan dissenyades específicament per a la reducció de soroll Gaussià i, llavors, no són capaços de reduir altres varietats de soroll com l'impulsiu o uniforme.

Tècniques de Filtratge per Substitució

Un tipus de soroll que contamina habitualment les imatges digitals és l'anomenat soroll impulsiu o *fat-tailed*. El soroll impulsiu afecta solament a certs píxels de la imatge mentre que els altres romanen inalterats.

Les tècniques de substitució ajuden treballant únicament als píxels sorollosos mantenint les característiques de la imatge (vèrtexs i detalls) inalterades. Quan les imatges es contaminen amb soroll impulsiu els esquemes

de substitució són extensament utilitzats a causa del seu rendiment suficient i una bona complexitat computacional provada. Els filtres amb esquemes de substitució fan servir diferents mètodes per identificar els impulsos. Per exemple, en [1] es realitza una anàlisi al grup o *cluster* format pel veïnat del píxel i detecta com a *soroll* aquells píxels el grau de pertinença al grup dels quals és baixa. El test del filtre de mediana vectorial *t-student* (tTVMF) [11] assumeix que els veïns del píxel processat segueixen alguna mena de distribució multi-normal. Es comprova per a cada component de color la pertinença a la distribució corresponent amb cert nivell de confiança. Si almenys un dels components no pertany a la distribució, llavors el píxel es considera sorollós. El treball en [39] fa servir un sistema d' inferència fuzzy que pren com entrades certes mesures estadístiques tant del píxel que està processant-se com del seu veïnat. El mètode en [62] comprova la diferència entre el vector d'entrada i la mitjana de certs vectors del veïnat amb baixa classificació d'un ordre donat. A [68] s'actua fent la detecció amb el vector d'entrada (píxel analitzat), la mediana vectorial, la mitjana vectorial i les distàncies acumulades als altres vectors de la finestra de filtratge. El treball en [69] amplia la forma de treball de [68] fent servir l'aproximació de la variància al cas multivariable. La solució presentada a [67] utilitza coeficients centrals ponderats i els mètodes de [73, 102, 104] fan servir un mètode d'ordenació de similitud vectorial per augmentar la importància del píxel considerat en el procés de detecció d'impulsos.

El concepte de *peer group* en [21, 37, 49] es fa servir per a detectar i filtrar soroll impulsiu. Els filtres introduïts per [21, 49] fan servir les diferències entre el *peer group* del píxel analitzat i altres *peer groups* dins del seu veïnat per a formar la regla de detecció. El treball de [38] proposa fer servir finestres de distint tamany per a determinar la *peer region* de cada píxel i llavors comprovar tant el tamany com la forma de la *peer region*. A la solució de [105], per a declarar un píxel com a *lliure de soroll* és condició necessària tenir un *peer group* d'una determinada cardinalitat al seu veïnat.

Com s'ha comentat en aquest apartat, els esquemes de substitució són molt útils per a processar imatges contaminades amb soroll impulsiu, donada la seua simplicitat computacional i el seus resultats positius. Això és relativament important, ja que en aquesta Tesi es treballa amb aquests esquemes i centrant-nos en soroll impulsiu. Per contra no són adequats per a altre tipus de soroll com per exemple el Gaussià.

Tècniques de Filtratge Fuzzy

Tenint en compte que les imatges són altament no estacionàries als vèrtexs i vist que és complicat distingir entre soroll i píxels que representen vèrtexs, els conjunts fuzzy, que són capaços de treballar amb incerteses o llindars, són molt apropiats per a tasques de filtratge d'imatges. De fet, la capacitat de gestionar les incerteses, cosa que és inherentment adaptable, implica que els filtres fuzzy són adequats per a l'eliminació de diferents tipus de soroll.

Al llarg dels últims anys s'han desenvolupat una gran quantitat de mètodes de reducció de soroll basat en models fuzzy per a imatges en tons de gris [50], filtre adaptable d'histograma fuzzy (HAF) [118, 119], mètode de detecció i reducció de soroll impulsiu fuzzy (FIDRM) [107], filtre basat en el control fuzzy iteratiu (IFCF) [27], filtre de substitució adaptable fuzzy (AFSF) [120], filtre de similitud fuzzy (FSB) [46], mètode de reducció de soroll impulsiu aleatori fuzzy i molt més. Aquests filtres fuzzy van ser desenvolupats amb imatges corruptes amb soroll *fat-tailed*, similar al soroll impulsiu. Fan servir enfocaments que milloren els esquemes dels filtres basats en ordenacions (com el VMF). Encara que aquests filtres estan especialment desenvolupats per a imatges en tonalitats de gris, poden fer-se servir en imatges en color aplicant-se per separat a cada component de color. Tot i això, aquest enfocament introdueix alguns *artefactes* de color principalment en vèrtexs i textures.

Certs modes de filtratge fuzzy per a imatges en color s'han proposat per a sobreposar-se a aquests problemes. Les operacions de mediana vectorial s'han estès a càlculs fuzzy en [17]. A [2] un sistema basat en regles fuzzy determina la sortida del filtre. En [39] un sistema d'inferència fuzzy (FISF) per a detectar impulsos en imatges en color es combina amb un esquema de substitució per a seleccionar com a sortida entre la identitat o la sortida d'un disseny de L-filtre proposat. Aquest L-filtre es dissenya per explotar les tècniques d'ordenació dels filtres de mediana vectorial. La sortida final es calcula fent servir la magnitud i direcció òptimes dels vectors. La mediana vectorial i certes mètriques fuzzy s'utilitzen en [97, 98, 99] per a calcular la sortida com una mitjana ponderada de les entrades. A [64, 65] els coeficients fuzzy determinen la sortida de l'algorisme seleccionant el vector d'entrada més representatiu o com la combinació de vectors dins de la finestra de filtratge. Un mètode fuzzy de detecció i eliminació de soroll impulsiu per a imatges en color alternatiu és el FIDRMC, estudiat a [108], que no utilitza tots els vectors completament. El resultat de la detecció, que s'aplica a cada component de color individualment, es fa servir per a calcular les diferències del color de les components lliures de soroll. Aquestes diferències es fan servir després per al mètode de reducció de soroll conservant les diferències de les components de color.

2.6 Avaluació Objectiva del Rendiment del Filtres d'Imatges en Color

En compte de fer l'avaluació visual d'una imatge, que és inherentment subjectiva, es necessiten algunes formes de mesurar o avaluar el rendiment d'una tècnica de filtratge des d'un punt de vista totalment objectiu. Un procediment utilitzat per avaluar objectivament el rendiment obtingut és veu a la Figura 2.1:

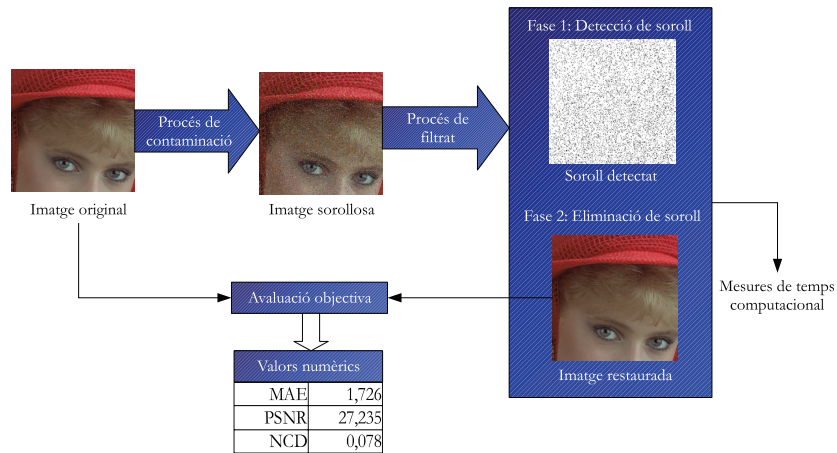


Figura 2.1. Esquema del procediment per avaluar objectivament un mètode particular de filtratge.

Primer, se seleccionen algunes imatges adequades per al processament. Les imatges han de presentar característiques interessants, com a vores fortes, detall subtils i/o àrees amb textures apropiades per als tests. Les Figures 2.2 i 2.3 mostra diverses imatges de test, algunes d'elles ben conegudes, que es fan servir en la comunitat científica i que s'han fet servir en aquesta tesi doctoral.

Segon, se simula la introducció d'un cert tipus de soroll a la imatge. Per això, necessitem un model de simulació de soroll. Principalment, es consideren els dos tipus de soroll amb que podem afectar les imatges en color. Per una part, el soroll relacionat amb el sensor de la càmera o calor tèrmic, i per altra part, el soroll que pot introduir-se durant la transmissió del senyal per un canal sorollós o contaminant [84].

El soroll associat al sensor de la càmera o soroll termal es modela com soroll Gaussià *blanc* tenint la següent distribució de probabilitat en cada canal de color:

$$p(x_n) = \frac{1}{(2\pi\sigma)^{\frac{1}{2}}} e^{-\frac{x^2}{2\sigma^2}} \quad (2.1)$$

on σ representa la desviació típica de la distribució. Aquest soroll s'introdueix en cada canal de color independentment, però tot i això, pot assumir-se que tots els tres canals de color tenen tant el mateix soroll mitjà com la mateixa variància de soroll respecte la imatge sencera.

El soroll per transmissió es modela habitualment com a soroll impulsiu. El procés de corrupció per soroll impulsiu sols afecta certs píxels de la imatge, deixant els altres inalterats. Típicament, el procés de contaminació canvia una o més components del píxel afectat reemplaçant els valors originals per valors que normalment són desviats dels originals. El model més comú de

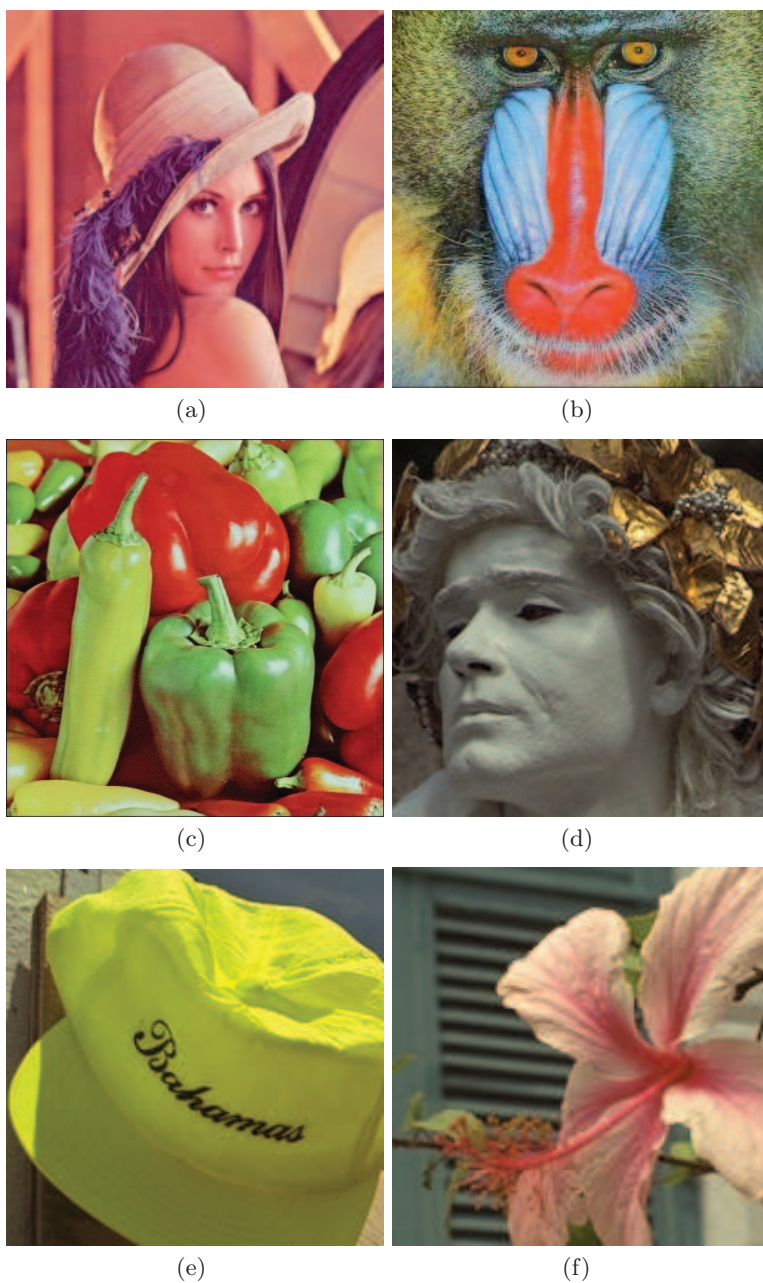


Figura 2.2. Algunes imatges clàssiques de test a l'avaluació de filtres: (a) Lenna (256×256), (b) Baboon (256×256), (c) Peppers (512×512), (d) Statue (200×200), (e) Caps (200×200), (f) Flower Window (150×150)

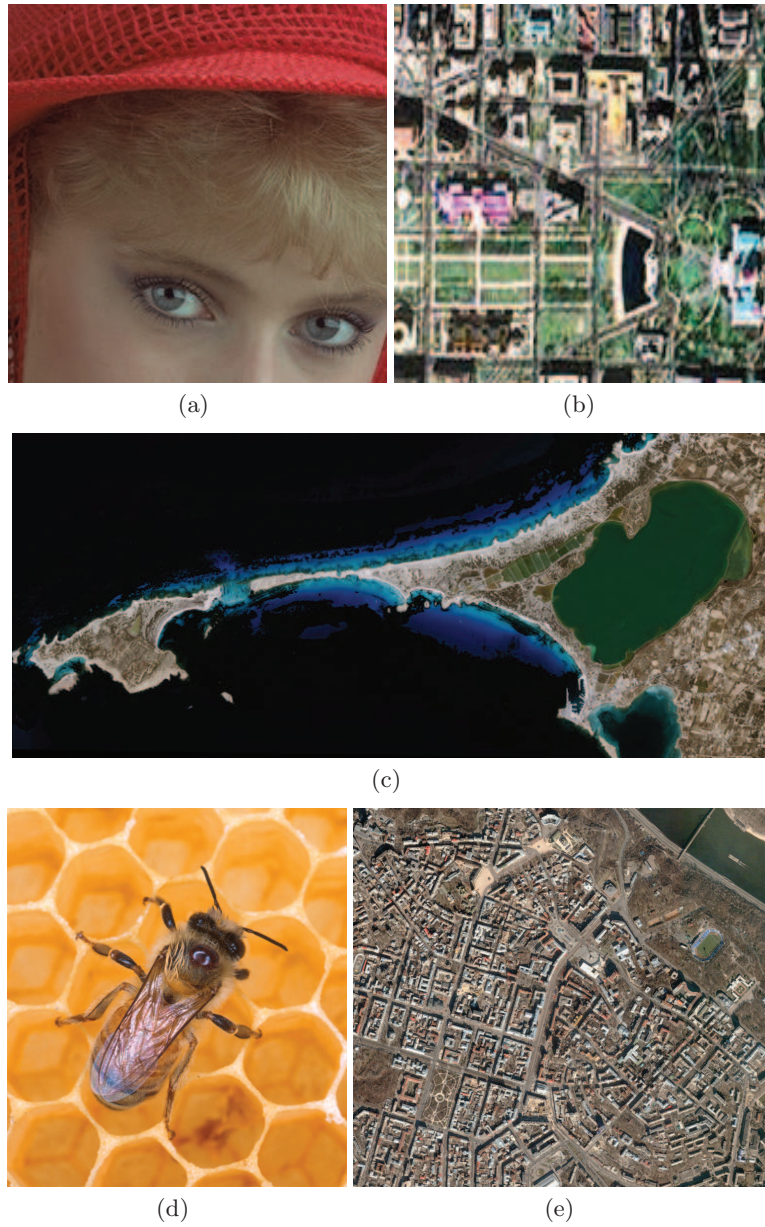


Figura 2.3. Algunes imatges clàssiques de test a l'avaluació de filtres: (a) Girl Red Hat (200×200), (b) Aerial (200×150), (c) Formentera (4168×9740), (d) Bee (2185×2440), (e) Kiev (3600×3240).

soroll impulsiu considera que l'impuls és o bé un valor extrem dins el rang del senyal o un valor aleatori uniforme distribuït dins del rang del senyal. Per a imatges RGB aquestes possibilitats es representen amb els següents models ben coneguts. A l'anomenat soroll impulsiu tipus I o model d'impulsos de valor fix, el procés de corrupció es modela com segueix:

$$\mathbf{F}^* = \begin{cases} \{d_1, F_G, F_B\} & \text{amb probabilitat } p \cdot p_1, \\ \{F_R, d_2, F_B\} & \text{amb probabilitat } p \cdot p_2, \\ \{F_R, F_G, d_3\} & \text{amb probabilitat } p \cdot p_3, \\ \{d_1, d_2, d_3\} & \text{amb probabilitat } p \cdot \left(1 - \sum_{i=1}^3 p_i\right). \end{cases} \quad (2.2)$$

on $\mathbf{F} = \{F_R, F_G, F_B\}$ representa el píxel original, \mathbf{F}^* representa el píxel corrupte pel procés, d_1, d_2, d_3 són valors independents iguals a 0 o 255 amb la mateixa probabilitat. El símbol p és la probabilitat que el píxel processat es transforme en corrupte i $p_i (i = 1, 2, 3)$ determina la probabilitat de en quin canal de color apareixerà el soroll.

A l'anomenat com a soroll impulsiu tipus II o model d'impulsos de valors aleatoris, $\mathbf{F}^* = \{d_1, d_2, d_3\}$ s'obté fent servir d_1, d_2, d_3 que seran valors enters independents distribuïts de manera uniforme a l'interval $[0, 255]$ amb probabilitat p .

Finalment, la imatge corrupta es filtra fent servir el procés de filtratge a ser avaluat i la imatge resultant es compara amb l'original lliure de soroll amb l'objectiu de mesurar el grau en el que la imatge de sortida és similar a la imatge original. S'han fet servir diferents funcions per a mesurar la similitud. En tal d'avaluar adequadament la qualitat del filtre s'han d'avaluar tant la eliminació del soroll com la conservació de detalls. L'*Error Mitjà Absolut* (MAE) és la funció més utilitzada per avaluar la conservació de detalls i el *Ratio de Pics de Soroll a la Senyal* (PSNR) és la funció feta servir per a expressar la capacitat en l'eliminació de soroll. Addicionalment, la *Diferència de Color Normalitzada* (NCD) es fa servir també molt, ja que mesura la percepció humana [105]. Aquestes tres mesures de qualitat o mesures de bondat del filtre s'han fet servir en aquesta Tesi i es defineixen com segueix [84]:

$$MAE = \frac{\sum_{i=1}^{N \cdot M} \sum_{q=1}^Q |F_i^q - \hat{F}_i^q|}{N \cdot M \cdot Q} \quad (2.3)$$

$$PSNR = 20 \log \left(\frac{255}{\sqrt{\frac{1}{NMQ} \sum_{i=1}^{N \cdot M} \sum_{q=1}^Q (F_i^q - \hat{F}_i^q)^2}} \right) \quad (2.4)$$

on M, N són les dimensions de la imatge, Q és el número de canals de la imatge ($Q = 3$ per a imatges en color), i F_i^q i \hat{F}_i^q representa la q -èsima component de la imatge original i filtrada al píxel i respectivament. Finalment

$$NCD_{Lab} = \frac{\sum_{i=1}^{N \cdot M} \Delta E_{Lab}}{\sum_{i=1}^{N \cdot M} E_{Lab}^*} \quad (2.5)$$

on $\Delta E_{Lab} = [(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2]^{\frac{1}{2}}$ representa l'error en la percepció del color i $E_{Lab}^* = [(L^*)^2 + (a^*)^2 + (b^*)^2]^{\frac{1}{2}}$ és la *norma* o *magnitud* del vector d'imatge en color a l'espai de color $L^*a^*b^*$. Aquest és un model de color normalitzat molt complet, fet servir convencionalment per descriure tots els colors visibles a l'ull humà. El paràmetre L indica la lluminositat ($0 = \text{negre}$ i $100 = \text{blanc}$), a indica la posició entre el verd i magenta ($a < 0 \leftrightarrow \text{verd}$ i $a > 0 \leftrightarrow \text{magenta}$) i el paràmetre b indica la posició entre el groc i el blau ($b < 0 \leftrightarrow \text{blau}$ i $b > 0 \leftrightarrow \text{groc}$).

Com que l'objectiu d'aquesta Tesi, a banda de seguir millorant el comportament dels filtres abans mencionats, és la millora computacional dels filtres, s'ha mesurat també el temps emprat en filtrar la imatge i s'ha comparat eixe temps amb els temps obtinguts dels altres filtres.

3 Fonaments de Conjunts Fuzzy: Lògica, Topologia i Mètriques

3.1 Concepte de Conjunt Fuzzy

En 1965, Lofti A. Zadeh [123] va definir el concepte de Conjunt Fuzzy (borrós o difús) com una classe d'objectes que es caracteritzen per una funció de pertinença que assigna a cada objecte un grau de pertinença entre 0 i 1. La motivació d'aquest concepte apareix perquè les classes d'objectes al món físic real no tenen en la majoria de les vegades, un criteri totalment definit de pertinença. Un conjunt a la teoria clàssica de conjunts sempre té un límit marcat perquè la pertinença a un conjunt és un concepte de blanc o negre, és a dir, un objecte o pertany completament al conjunt o no pertany completament.

La teoria de conjunts fuzzy és una generalització de la teoria clàssica de conjunts que permet la pertinença parcial d'un element a un conjunt, ja que d'aquesta manera, podem descriure una transició suau i gradual entre les regions fora del conjunt a aquelles que el conjunt representa. Aquestes classes d'objectes no són conjunts en el sentit clàssic, però juguen un paper important al domini del reconeixement de formes, comunicació de la informació i abstracció.

Per referir-nos a un conjunt fuzzy generalment farem servir l'estructura lingüística “ x és P ”, on x és un element de l'Univers del Discurs i P és un predicat. Així, direm que “ X és molt alt” admetent l'existència del conjunt fuzzy dels homes alts i apareix la variable lingüística *ALTURA* que pren valors lingüístics o etiquetes: “molt alt”, “alt”, Un home d'1.85m podem considerar que pertany a ambdós conjunts, o que satisfà el dos predicats, amb grau d'acompliment de, per exemple, 0.6 i 0.8, respectivament.

Basada en la teoria de conjunts fuzzy apareix la *Lògica Fuzzy* que modela els predicats amb un desenvolupament anàleg a la dualitat existent entre la teoria de conjunts de Cantor i la lògica binària, fent servir el concepte de *t-norma*.

La Lògica Fuzzy és una forma de raonament que permet incorporar als sistemes d'automatització esquemes de raonament típicament humans, apareixent per tal de tractar el coneixement imprecís, bàsicament al context de la construcció de Sistemes Experts encara que la seua major aplicació en l'actualitat són els controls de processos. Altres grans camps d'aplicació són:

- Sistemes d'Informació Intel·ligents.

- Reconeixements de Formes.
- Tractament de Senyals.
- Robòtica.

Certament en poques ocasions de la Història de la Ciència un concepte matemàtic ha tingut una eclosió tan espectacular en diversos camps de les ciències: Matemàtiques, Enginyeria, Biologia, Economia,...

L'èxit de la Lògica Fuzzy és que permet treballar als ordinadors no solament amb mètodes quantitativs sinó qualitativs. Pot tractar-se informació amb límits imprecisos, graduacions, de qualsevol format d'entrada, procés, càlcul, memòria o sortida. Amb tot això s'intenta elevar les funcions i capacitats de la màquina a nivells comparables als del ésser humà, el que habitualment anomenem com a Intel·ligència Artificial.

El desenvolupament de la teoria de conjunts fuzzy inclou conceptes que manifesten les relacions entre els conjunts clàssics i fuzzy, operacions en conjunts fuzzy, tipus i disseny de funcions de pertinença, propietats dels conjunts fuzzy i altres (veure [51, 115, 121]).

3.2 Principis de Lògica Fuzzy

El terme de lògica fuzzy es fa servir en dos vessants distintes. En sentit estricta, la lògica fuzzy fa referència a sistemes lògics que generalitzen la lògica bi-valuada per a raonaments sota certes condicions. En un sentit ample, la lògica fuzzy es refereix a totes les teories i tecnologies que fan servir lògica fuzzy. A tot això, dins d'aquest sentit ample, podem explicar les bases de la lògica fuzzy amb tres conceptes bàsics:

1. Els conjunt fuzzy abans mencionats
2. Variables lingüístiques
3. Regles *if-then* (si-llavors)

Variables Lingüístiques

Havent introduït els conceptes fonamentals dels conjunts fuzzy, és natural veure com poden fer-se servir. A l'igual que un conjunt convencional, un conjunt fuzzy pot fer-se servir per a descriure el valor d'una variable. Per exemple la sentència "L'edat d'una persona adulta" fa servir el conjunt "Persona Adulta" per descriure la quantitat d'anys d'una persona. Més formalment pot descriure's com *QuantitatEdat és Adulta*. La variable *QuantitatEdat* és un exemple que demostra un concepte important dins la lògica fuzzy: la variable lingüística. Una variable lingüística permet que el seu valor s'expressi tant de forma qualitativa mitjançant termes lingüístics (això és, un símbol del conjunt fuzzy) com de forma quantitativa (expressant el grau de pertinença al conjunt fuzzy). El terme lingüístic es fa servir per a expressar conceptes i

coneixement humà a les comunicacions, mentre que la funció de pertinença és útil per a processar dades d'entrada numèriques. Per exemple, en la figura 3.1 s'ha elegit representacions trapezoïdals, però també és bastant usual l'ús de figures triangulars.

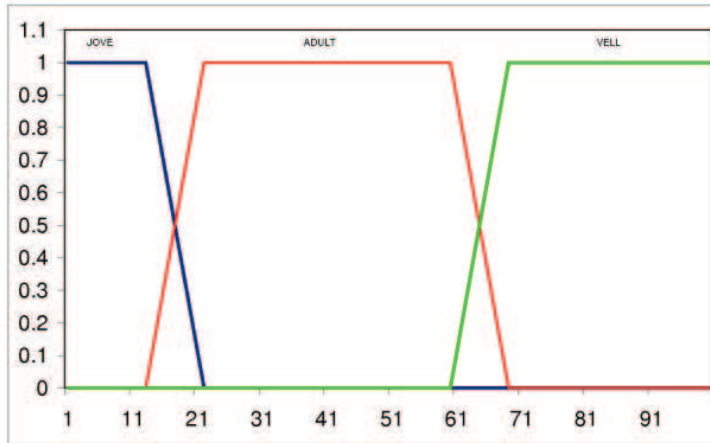


Figura 3.1. Representació de *QuantitatEdat* mitjançant conjunts fuzzy.

Una variable lingüística és com la composició d'una variable simbòlica (una variable el valor de la qual és un símbol) i una variable numèrica (una variable el valor del qual és un número). Al nostre exemple de les edats, hi ha altres descripcions lingüístiques com “jove”, “adult”, “vell” i d'altres possibles. Totes aquestes descripcions lingüístiques, que de fet són vagues i imprecises, poden representar-se mitjançant conjunts fuzzy. D'aquesta forma, el valor numèric de la variable *QuantitatEdat* s'expressa en termes del seu grau de pertinença als conjunts fuzzy representats. La figura 3.1 mostra un exemple de representació de la descripció lingüística de la variable *QuantitatEdat* mitjançant conjunts fuzzy.

Regles Fuzzy *if-then*

Al llarg de totes les tècniques desenvolupades per als conjunts fuzzy, la regla *if-then* fuzzy (abreujadament regla fuzzy) és de llarg la més clara per l'ample ventall d'aplicacions. Les regles fuzzy s'han aplicat a diverses disciplines com a sistemes de control, presa de decisions, reconeixement de formes i modelat

de sistemes [18, 115, 121]. Les regles fuzzy juguen un paper molt important en aplicacions industrials anant des de productes de consum, robòtica, manufactures, control de processos, imatges mèdiques fins a gestions comercials i financeres.

La inferència basada en regles fuzzy pot entendre's des de distints punts de vista. Conceptualment, pot entendre's fent servir la metàfora de dibuixar una conclusió amb un grup d'experts. Matemàticament, es pot veure com un esquema d'interpolació. Formalment, és una generalització del model d'inferència lògica *modus ponens*.

Dins la lògica clàssica, si coneixem que una regla és vertadera i també coneixem que l'antecedent de la regla és vertader, podem inferir pel *modus ponens* que el conseqüent també és vertader. Per exemple, suposem que la següent regla R1 és certa:

Regla 1 R1:

IF *el salari anual d'una persona és major que 120000 Euros*
THEN *la persona és rica*

També sabem que la següent sentència és certa: *El salari anual de Maria són 121000 Euros.*

Basant-se en el *modus ponens*, la lògica clàssica pot deduir que la següent sentència és vertadera *Maria és rica*

Una limitació del *modus ponens* és que no pot distingir coincidències parcials. Per a mostrar-ho, considerem la regla R1, i el cas d'una persona que guanya 119000 Euros. Tothom diria que aquesta persona és *una mica rica*, per contra el *modus ponens* ni deduirà que eixa persona és *rica* ni tampoc *no rica*. El problema és per dos motius: (1) l'antecedent de R1 no representa una transició suau a la categoria rica que habitualment si està present al raonament humà, i (2) el *modus ponens* no pot distingir una situació en la que l'antecedent d'una regla és parcialment satisfactori. Tenint en compte aquesta limitació, les regles d'inferència fuzzy generalitzen el *modus ponens* per a permetre que la conclusió inferida siga modificada segons el grau en el que l'antecedent se satisfà. Aquesta és l'essència de la regles d'inferència fuzzy.

Anàlogament al cas de la lògica clàssica, l'estructura d'una regla fuzzy té dos components, la part del *if* (referida com a antecedent) i la part del *then* (coneguda com conseqüent).

Regla Fuzzy 1 Estructura d'una regla fuzzy:

IF *<antecedent>*
THEN *<conseqüent>*

Ja que l'objectiu del present text no és entrar en els detalls de les regles fuzzy així com els sistemes basats en regles fuzzy, a continuació, es donarà una explicació breu i informal de les bases dels procediments de la forma de

treball amb regles fuzzy. El lector curiós pot trobar extensa informació a la literatura, per exemple a [115, 121, 123].

D'acord amb l'anterior, la lògica clàssica, amb regles clàssiques, pot determinar quan el conseqüent d'una regla se satisfà o no mirant si l'antecedent s'ha satisfet o no. Al cas de la lògica fuzzy, mitjançant regles fuzzy on cert coneixement del grau en el que l'antecedent se satisfà, el grau en el que el conseqüent se satisfà pot calcular-se mitjançant un procés anomenat *inferència basada en regles fuzzy*. Qualsevol sistema basat en regles fuzzy fa servir diverses regles. El mode en què els sistemes basats en regles fuzzy treballen pot veure's de manera simple com segueix. Anem a assumir que el sistema basat en regles fuzzy està format per un conjunt de variables d'entrada, un conjunt de regles fuzzy i un conjunt de variables de sortida els valors de les quals han de ser calculades pel sistema. El mecanisme de procés del sistema es divideix en tres fases:

1. Partint de la base dels valors numèrics de les variables d'entrada, es calcula el grau en el qual els antecedents de les regles fuzzy s'acompleixen. Aquesta fase s'anomena com *fuzzification*.
2. El procés d'inferència basat en regles fuzzy s'aplica per a calcular el grau en què els conseqüents se satisfan.
3. Finalment, les variables de sortida, que no han de ser necessàriament fuzzy, es calculen amb el grau en què s'han satisfet els conseqüents amb les regles fuzzy. Aquesta fase s'anomena *defuzzification*.

3.3 Principis de Topologia i Mètrica Fuzzy

A continuació, s'explica el concepte de mètrica fuzzy dins del context de la teoria fuzzy.

Un dels primer conceptes recercats a les matemàtiques fuzzy va ser la topologia fuzzy. El primer treball en topologia fuzzy el va fer C. L. Chang en 1968 [19]. Segons Chang, una topologia τ en X és una família de conjunts fuzzy en X que és tancat per unions qualssevol i per interseccions finites. A més a més aquesta família ha de contenir les funcions constants 0 i 1. Noteu que aquest és el concepte més utilitzat a la literatura existent però la teoria desenvolupada per contra no és l'única. Des d'altre punt de vista, R. Lowen [57, 58] exigeix, com ampliació, que τ ha de contenir totes les funcions constants. Aquest tipus de topologia s'anomena *topologia laminada* [100]. Per tant, a diferència de la topologia de Chang, una topologia laminada no constitueix una generalització de la topologia en el sentit clàssic. Per altra part, Goguen [32] i Hutton [42] generalitzen el concepte donat per Chang reemplaçant el rang fuzzy $[0, 1]$ per un reticle complementat L , apareixent el concepte de L -topologia. Un dels problemes més interessant i estudiat en topologia fuzzy és obtenir una noció apropiada d'espai mètric fuzzy. Cal recordar que l'estudi d'espais mètrics es basa en la noció de distància entre punts, tot i això, en

diverses situacions reals aquesta distància no pot determinar-se exactament. Aquest problema, que pertany a l'àrea fuzzy, es va abordar prèviament des del punt de vista de la teoria de la probabilitat. De fet, en 1942 K. Menger [78] va introduir els anomenats espais mètrics probabilístics. En aquests espais, si $d(x, y)$ és la distància entre dos punts x i y llavors la funció de distribució $F_{xy}(t)$ representa la probabilitat que la distància entre x i y siga menor o igual a t . Posteriorment, Schweizer i Sklar [95, 96], van seguir l'estudi d'aquests espais i recentment han estat publicats altres treballs al voltant d'aquesta vessant [15], [80], [81], [86], [95], [111].

És fàcil adonar-se'n que la noció de topologia fuzzy ha estat estudiada des de distints punts de vista a l'igual que en quasi tots els conceptes fuzzy estudiats. Quant a les mètriques fuzzy, també diversos autors han estudiat aquest concepte des de diferents punts de vista. Ací farem una classificació d'aquests treballs en dos grans grups: per una banda un primer grup format per aquells treballs on una (pseudo-)mètrica fuzzy en X es tracta com una funció $d : \Omega \times \Omega \rightarrow \mathbb{R}$ on $\Omega \subset I^X$ ($I = [0, 1]$) que satisfà certs axiomes *anàlegs* a altres dins les mètriques clàssiques. Entre aquests treballs destaquem els de Deng Zi-ke [22], Erceg [26], Hu [41], i Artico i Moresco [3]. Els problemes més interessants en aquesta línia són:

1. Investigar de quina forma una mètrica fuzzy produeix una quasi-uniformitat en el sentit de [43] i una topologia fuzzy [26, 22, 41].
2. Determinar el criteri de (pseudo-)medició [25, 26, 41].
3. Definir les propietats de la disjunció en espais mètrics [26, 3, 41, 23].
4. Definir les propietats de completació i acotació [3, 23].

Per altra part, un segon grup inclou aquells treballs on la distància entre objectes és fuzzy. El resultat més destacats en aquest aspecte es deuen a Kaleva i Seikkala [48] i Kramosil i Michalek [55].

Al nostre treball, farem servir el concepte de mètrica fuzzy proporcionat per George i Veeramani. Podem definir aquest concepte com una modificació *adequada* de la introduïda per Kramosil i Michalek [55].

A continuació farem un resum cronològic per mostrar els resultats més importants de la teoria de mètriques fuzzy.

3.4 Espais Mètrics Probabilístics

Anteriorment a la introducció de la teoria fuzzy dins el camp dels espais mètrics, l'estudi de mètriques estava associat a conceptes probabilístics. En aquest sentit, en 1942 Menger [78] va definir el concepte d'espai mètric probabilístic com segueix.

Definició 3.4.1 *Siga X un conjunt arbitrari no buit. Siga F_{pq} una família de funcions de distribució que satisfan el següent:*

$$(M1) \quad F_{pq}(0) = 0$$

- (M2) Si $p = q$, llavors $F_{pq}(x) = 1 \forall x > 0$
 (M3) Si $p \neq q$, llavors $F_{pq}(x) < 1$ per a algun $x > 0$
 (M4) $F_{pq} = F_{qp}$
 (M5) $F_{pr}(x+y) \geq T(F_{pq}(x), F_{qr}(y)) \forall p, q, r \in X$ i $\forall x, y \in \mathbb{R}$, en $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ és una funció que satisfà:

- (i) $T(a, b) = T(b, a)$
 (ii) $T(a, b) \leq T(c, d)$ si $a \leq c$ y $b \leq d$
 (iii) $T(a, 1) > 0$ si $a > 0$, i $T(1, 1) = 1$

La mètrica estadística $F_{pq}(x)$ pot interpretar-se com la probabilitat que la distància entre dos punts p i q siga menor que x .

Schweizer i Sklar [96] substitueixen la condició anterior (M5) per la següent:

$$\text{Si } F_{pq}(x) = 1 \text{ i } F_{qr}(y) = 1, \text{ llavors } F_{pr}(x+y) = 1$$

L'espai resultant s'anomena espai mètric probabilístic dèbil i és una generalització dels espais probabilístics de Menger.

Schweizer i Sklar, també introduïren el concepte de t -norma contínua que té un important pes dins del desenvolupament de la teoria d'espais mètrics fuzzy.

Definició 3.4.2 Una operació binària $* : [0, 1] \times [0, 1] \rightarrow [0, 1]$ és una t -norma contínua si satisfà les següents condicions:

- (i) $*$ és contínua, associativa i commutativa
 (ii) $a * 1 = a \forall a \in [0, 1]$
 (iii) $a * b \leq c * d$ si $a \leq c$ i $b \leq d$, $a, b, c, d \in [0, 1]$

Amb base la definició anterior, Schweizer i Sklar van definir un espai mètric probabilístic com segueix.

Definició 3.4.3 Un espai mètric probabilístic és un parell (X, F) on X és un conjunt arbitrari i F és una aplicació de $X \times X$ al conjunt de totes les funcions de distribució que satisfan:

- (1) $F_{xy}(t) = 1 \forall t > 0$ si i sols si $x = y$
 (2) $F_{xy}(0) = 0$
 (3) $F_{xy} = F_{yx}$
 (4) Si $F_{xy}(t) = 1$ i $F_{yz}(s) = 1$, llavors $F_{xz}(t+s) = 1$

Un espai de Menger $(X, F, *)$ és un espai mètric probabilístic junt amb una t -norma si satisfà la condició

$$F_{xz}(t+s) \geq F_{xy}(t) * F_{yz}(s)$$

3.5 Espais Mètrics Fuzzy de Kaleva i Seikkala

Atès que la incertesa entre la distància existent entre dos punts està més relacionada a la noció fuzzy que a l'aleatorietat, Kaleva i Seikkala [48] ampliaren el concepte d'espai mètric a la nova teoria fuzzy associant la distància entre dos punts a un número fuzzy.

Definició 3.5.1 *Un número fuzzy és una aplicació $x : \mathbb{R} \rightarrow [0, 1]$ que associa el grau de pertinença de cada número real.*

Un número fuzzy es diu que és convex si $x(t) \geq \min\{x(s), x(r)\}$ quan $s \leq t \leq r$.

Per a $0 < \alpha \leq 1$ i un número fuzzy x , els conjunts de α -nivell $[x]_\alpha$ es defineixen per

$$[x]_\alpha = \{u : x(u) \geq \alpha\}$$

Com a resultat, x és convex si i sols si $[x]_\alpha$ és un conjunt convex en $\mathbb{R} \forall \alpha \in]0, 1]$

Adicionalment, si existeix un element $u \in \mathbb{R}$ tal que $x(u) = 1$, llavors el número fuzzy x és *normal*.

Un número fuzzy es diu que és no-negatiu si $x(u) = 0 \forall u < 0$.

El conjunt de tots els números fuzzy semi-continus no negatius es denota per \mathbb{G} .

Fent servir la notació de Kaleva i Seikkala definim el concepte d'espai mètric fuzzy com segueix:

Definició 3.5.2 *Siga X un conjunt arbitrari no buit i siga $d : X \times X \rightarrow \mathbb{G}$ una aplicació. Siguen $L, R : [0, 1] \times [0, 1] \rightarrow [0, 1]$ dos aplicacions simètriques no decreixents en ambdós sentits que satisfan $L(0, 0) = 0$ i $R(1, 1) = 1$. Denotem per*

$$[d(x, y)]_\alpha = [\lambda_\alpha(x, y), \rho_\alpha(x, y)] \text{ per a } x, y \in X, 0 < \alpha \leq 1$$

La cuaterna (X, d, L, R) s'anomena espai mètric fuzzy KS i d és una mètrica fuzzy KS si s'acompleixen les condicions següents:

- (a) $d(x, y) = 0$ si i sols si $x = y$
- (b) $d(x, y) = d(y, x)$ per algun $x, y \in X$
- (c) per algun $x, y, z \in X$,
 - (1) $d(x, y)(s + t) \geq L(d(x, z)(s), d(z, y)(t))$
si $s \leq \lambda_1(x, z)$, $t \leq \lambda_1(z, y)$ i $s + t \leq \lambda_1(x, y)$
 - (2) $d(x, y)(s + t) \leq R(d(x, z)(s), d(z, y)(t))$
si $s \geq \lambda_1(x, z)$, $t \geq \lambda_1(z, y)$ i $s + t \geq \lambda_1(x, y)$

Ja que els números reals no negatius pertanyen a \mathbb{G} , si assumim que

$$L(a, b) = 0 \quad i \quad R(a, b) = \begin{cases} 0 & a = b = 0 \\ 1 & \text{en altre cas} \end{cases}$$

llavors l'espai mètric habitual pot considerar-se com un espai mètric fuzzy *KS*. Addicionalment, la definició 3.5.2 generalitza els espais de Menger com es prova a la nota següent.

Nota 3.5.3 *Siga $(X, F, *)$ un espai de Menger. Definim $d : X \times X \rightarrow \mathbb{G}$ com a:*

$$d(x, y)(t) = \begin{cases} 0 & t < t_{xy} = \sup\{t : F_{xy}(t) = 0\} \\ 1 - F_{xy}(t) & t \geq t_{xy} \end{cases}$$

*Si prenem $R(a, b) = 1 - ((1 - a) * (1 - b))$ i $L(a, b) = 0$, llavors (X, d, L, R) és un espai mètric fuzzy *KS* i per tant, un espai de Menger pot considerar-se com un espai mètric fuzzy *KS**

*En el cas que s'acomplisquen completament aquestes condicions, llavors $(X, F, *)$ és un espai de Menger on $a * b = 1 - R(1 - a, 1 - b)$ per a qualssevol $a, b \in [0, 1], x, y \in X, s \in \mathbb{R}$,*

$$F_{xy}(s) = \begin{cases} 0 & s \leq \lambda_1(x, y) \\ 1 - d(x, y)(s) & s \geq \lambda_1(x, y) \end{cases}$$

*A més a més, $(X, F, *)$ es coneix com l'espai de Menger associat.*

3.6 Espais Mètrics Fuzzy de Kramosil i Michalek

Kramosil i Michalek [55] van definir el concepte d'espai mètric fuzzy generalitzant el concepte d'espai mètric probabilístic a la teoria fuzzy de la manera següent:

Definició 3.6.1 [35, 55]

*Una terna $(X, M, *)$ es diu que és un espai mètric fuzzy de Kramosil i Michalek (espai mètric fuzzy *KM*) si X és un conjunt arbitrari, $*$ és una t -norma contínua i M és un conjunt fuzzy de $X \times X \times [0, +\infty[$ que satisfà les següents condicions per a qualssevol $x, y, z \in X$ i $t, s > 0$:*

- (KM1) $M(x, y, 0) = 0$
- (KM2) $M(x, y, t) = 1 \quad \forall t > 0$ si i sols si $x = y$
- (KM3) $M(x, y, t) = M(y, x, t)$
- (KM4) $M(x, y, t) * M(y, z, s) \leq M(x, z, t + s)$
- (KM5) $M(x, y, \cdot) : [0, +\infty[\rightarrow [0, 1]$ és contínua per la dreta

3.7 Espais Mètrics Fuzzy de George i Veeramani

El concepte de mètrica fuzzy introduïda per George i Veeramani [30, 36] és el que fem servir a la present dissertació. El concepte es defineix com una modificació de l'introduït per Kramosil i Michalek com segueix.

La terna $(X, M, *)$ és un espai mètric fuzzy si X és un conjunt arbitrari no buit, $*$ és una t -norma contínua i M és un conjunt fuzzy de $X \times X \times]0, +\infty[$ que satisfà els següents axiomes per a qualssevol $x, y, z \in X$, $t, s > 0$:

- (GV1) $M(x, y, t) > 0$
- (GV2) $M(x, y, t) = 1$ si i sols si $x = y$
- (GV3) $M(x, y, t) = M(y, x, t)$
- (GV4) $M(x, y, t) * M(y, z, s) \leq M(x, z, t + s)$
- (GV5) $M(x, y, \cdot) :]0, +\infty[\rightarrow]0, 1]$ és contínua

En el següent, entendrem per espai mètric fuzzy els de George i Veeramani. Com és lògic, ens referirem a un espai mètric fuzzy X sense anomenar explícitament la mètrica fuzzy si no és necessari.

La mètrica fuzzy M genera una topologia τ_M a X . La topologia τ_M té com a base la família de *boles obertes* $\{B_M(x, r, t) : x \in X, 0 < r < 1, t > 0\}$ on $B_M(x, r, t) = \{y \in X : M(x, y, t) > 1 - r\}$.

Una successió $\{x_n\}_{n=1}^{\infty}$ a X s'anomena una successió de Cauchy si per a cada $\varepsilon \in]0, 1[$ i $t > 0$, existeix $n_0 \in \mathbb{N}$ tal que $M(x_n, x_m, t) > 1 - \varepsilon$ si $m, n \geq n_0$. X es diu complet si cada successió de Cauchy és convergent. X és F -acotada si existeix $r \in]0, 1[$ tal que $M(x, y, t) > 1 - r$ per a qualssevol $x, y \in X$, $t > 0$.

Si (X, d) és un espai mètric, llavors la funció $M_d(x, y, t) = \frac{t}{t + d(x, y)}$ és una mètrica fuzzy (anomenada estàndard) en X , amb el producte usual com a t -norma contínua. La topologia τ_{M_d} coincideix amb la topologia induïda per d .

Una mètrica fuzzy $(M, *)$ en X es diu que és estacionària si M no depèn de t , això és, per a qualssevol $x, y \in X$ la funció $M_{x,y}(t) = M(x, y, t)$ és constant [34].

Un subconjunt A de X es diu que és F -acotat [30] si existeix $t > 0$ i $s \in]0, 1[$ tal que $M(x, y, t) > s$ per a tot $x, y \in A$.

Les definicions anteriors compleixen la teoria clàssica d'espais mètrics en el sentit que (X, d) és complet (acotat) si i sols si (X, M_d, \cdot) és complet (F -acotat).

Aquesta definició de mètrica fuzzy adequada mereix una atenció especial ja que, com varen demostrar Gregori i Romaguera [33] la classe d'espais *metrizables* coincideix amb els espais *metrizables fuzzy* de George i Veeramani.

$M(x, y, t)$ pot interpretar-se com el grau de proximitat entre x i y respecte a t . En eixe cas, d'acord a (GV2), $M(x, y, t) = 0$ ha d'associar-se al concepte ∞ .

Les tres t -normes més conegudes, que podem denotar-les per T_i ($i = 1, 2, 3$), són les següents:

$$T_1(x, y) = \min\{x, y\}$$

$$T_2(x, y) = xy$$

$$T_3(x, y) = \max\{0, x + y - 1\}$$

Part II

Algorismes Implementats

4 Detecció i Eliminació Ràpida de Soroll Impulsiu amb Peer Groups i Mètriques Fuzzy

Joan-Gerard Camarena, V. Gregori, S. Morillas, A. Sapena, Fast detection and removal of impulsive noise using peer groups and fuzzy metrics, *Journal of Visual Communication and Image Representation*, 19 (2008) 20–29.

Resum

En aquest capítol s'introdueix una idea novell per a la detecció de soroll en imatges en color. El concepte de *peer group* es redefineix fent servir una mètrica fuzzy. Aquesta idea es fa servir per a la detecció ràpida de píxels sorollosos aprofitant els avantatges de les propietats de la mètrica fuzzy. Basant-se en aquesta detecció de píxels sorollosos, es proposa un esquema de filtratge per substitució entre el filtre de mitjana vectorial (AMF) i l'operador d'identitat. L'esquema de filtratge per substitució obté un bon balanç entre l'eliminació del soroll i la conservació dels detalls de la imatge, sent més ràpid que els filtres de substitució més recents basats en el concepte de *peer group*.

4.1 Introducció

Durant l'adquisició de les imatges i el procés de transmissió, la qualitat de les imatges digitals pot veure's afectada per la introducció de soroll, difuminacions (*blurring*) o deformacions a causa de certes deficiències en l'adquisició i/o procés de transmissió, moviment a la càmera o desenfocament. Concretament, és habitualment en el procés de transmissió de dades on apareix el que es coneix com a soroll impulsiu. El procés de corrupció per soroll impulsiu sols afecta a certs píxels de la imatge, deixant els altres inalterats. Típicament, el procés de contaminació canvia una o més components del píxel afectat reemplaçant els valors originals per valors que normalment són desviats dels originals. El model més comú de soroll impulsiu considera que l'impuls és o bé un valor extrem dins el rang del senyal o un valor aleatori uniforme distribuït dins del rang del senyal. Dins aquest context, el procés de filtratge és una tasca essencial que ha d'aplicar-se per evitar possibles inconvenients en processos i tractaments posteriors de les imatges.

Fins la data s'han proposat diversos filtres vectorials per a imatges en color que prenen avantatge de la correlació existent entre els canals de colors . Una família coneguda de filtres vectorials no lineals es basen en la teoria

estadística robusta [20, 40]. Quan els vectors dins d'una finestra corredera predeterminada s'ordenen fent servir el principi d'ordenació reduït, els vectors amb menor rang són aquells que són *propers* a tots els altres vectors en la finestra, d'acord a la *distància o mesura de similitud* utilitzada [63, 84]. Per altra part, vectors atípics, susceptibles de ser declarats com a soroll o *outliers*, ocupen els rangs més alts. Llavors, en aquests filtres, la sortida del filtre normalment es defineix com el menor vector del rang. El *filtre de mediana vectorial* (VMF) [4], el *filtre direccional vectorial bàsic* (BVDF), [113], i el *filtre de distància direccional* (DDF), [47], són tècniques de filtratge ben conegudes d'aquesta família que obtenen un bon nivell d'eliminació de soroll i altres propietats interessants¹. El procés direccional d'imatges RGB s'introduí en tal de millorar el rendiment del VMF des del punt de vista de la cromaticitat [113]. Un altre filtre cromàtic pot trobar-se a [60]. Per altra part, la mitjana i la mediana vectorial poden estendre's a números fuzzy (difusos) a [17] per mitjà de certes distàncies fuzzy.

Els filtres vectorials tradicionals descrits anteriorment tenen el desavantatge que estan dissenyats per a realitzar una quantitat fixa de suavitzat (*smooth*) o filtratge i no són capaços d'adaptar-se a les característiques locals de la imatge [63]. S'han introduït recentment en la literatura diverses aproximacions amb l'objectiu de poder adaptar la suavització en cada cas particular. En distintes tècniques es calculen coeficients per a determinar la sortida del filtre com una combinació dels vectors en la finestra de filtratge. Esquemes que fan servir pesos per a operacions de medianes vectorials ponderades es proposen en [59, 61, 66, 70]. En [1, 11, 62, 68, 69, 105] els autors proposen primer si el vector analitzat pot considerar-se sorollós fent servir anàlisi de classes [1], un test respecte a la mitjana vectorial dels vectors de la finestra [62], un estudi estadístic respecte a la mitjana o mediana dels vectors [11, 68, 69], i després aplicar l'operació de filtratge solament a aquells píxels sorollosos. De forma similar, en [116] es fa servir un algorisme genètic per a decidir en cada posició de la imatge si fer l'operació del VMF, del BVDF o l'operació identitat. L'ús simultani de distints filtres vectorials en cada zona de la imatge i la combinació de les seues sortides particulars mitjançant funcions racionals per a determinar la sortida final és la idea proposada en [53]. A [67, 73, 88, 101, 102, 103, 104], es proposa prevaldre el píxel central en cada finestra de filtratge per tal de reduir el nombre de substitucions innecessàries.

El concepte de *peer group* amb la distància Euclídea clàssica s'utilitza en [105] per a detectar els píxels sorollosos impulsius. Altres distàncies clàssiques ben conegudes o mesures de similitud que poden fer-se servir en compte de la distància Euclídea per a determinar el *peer group* es revisen en [63]. En aquest capítol el concepte de *peer group* s'adapta per a l'ús d'una nova mètrica fuzzy. El fet d'usar la mètrica fuzzy ha estat donat perquè s'ha provat que és una alternativa adequada a diverses mètriques clàssiques [88] i així van introduint-

¹ Per a una descripció detallada d'aquests filtres vectorials, el lector pot consultar [63, 84]. Una descripció succinta es troba en aquesta Tesi Doctoral al Capítol 2

se dins d'aquesta àrea d'investigació. Aquesta mètrica fuzzy pot ser també interessant en altres tècniques de procés d'imatges basades en tècniques fuzzy. Les tècniques fuzzy són interessants dins del processament d'imatges ja que permeten incorporar el coneixement i la intuïció humanes a un disseny que no pot modelar-se amb la matemàtica clàssica [64]. A més a més, donades les no homogeneïtats de les imatges així com les dificultats per distingir entre soroll i vèrtexs, el modelat fuzzy es considera prou convenient per al filtratge d'imatges [64].

El concepte de *peer group* que definim s'utilitza per a dissenyar un filtre de substitució com a base del treball mencionat. El filtre del capítol actual és més ràpid que el proposat a [105], presentant un rendiment molt competitiu respecte a les tècniques de filtratge per substitució introduïdes recentment i millorant-les en certs casos. Els estudis experimentals amb la comparació del rendiments es presenten en tal de mostrar que el model proposat millora els filtres vectorials clàssics de manera eficient.

El capítol està organitzat de la següent forma. A la Secció 2 es presenta la mètrica fuzzy que fem servir. Dins la Secció 3 es detalla l'algorisme de filtratge proposat. Els estudis experimentals així com algunes comparatives davant treballs recents i filtres vectorials coneguts es veu a la Secció 4. A la Secció 5 s'estudia la complexitat computacional. Finalment a la Secció 6 veurem les conclusions.

4.2 Una Mètrica Fuzzy Adequada

Denotem $(x_i(1), x_i(2), x_i(3))$ el vector de color d'un píxel \mathbf{x}_i a l'espai de color RGB, i siga X el conjunt $\{0, 1, \dots, 255\}^3$ i fixat $K > 0$. Llavors, segons [88, 93], la funció $M : X \times X \rightarrow]0, 1]$ donada per

$$M(\mathbf{x}_i, \mathbf{x}_j) = \prod_{l=1}^3 \frac{\min\{x_i(l), x_j(l)\} + K}{\max\{x_i(l), x_j(l)\} + K} \quad (4.1)$$

és una mètrica fuzzy estacionària, per al producte usual, en X en el sentit de George i Veeramani. D'aquesta forma en aquest capítol considerarem $M(\mathbf{x}_i, \mathbf{x}_j)$ com la distància fuzzy entre els vectors de la imatge en color \mathbf{x}_i i \mathbf{x}_j . Òbviament M és F -acotada i satisfà

$$0 < \left(\frac{K}{255 + K} \right)^3 \leq M(\mathbf{x}_i, \mathbf{x}_j) \leq 1 \quad (4.2)$$

per a tot $\mathbf{x}_i, \mathbf{x}_j \in X$.

El proper estudi es basa en la mètrica fuzzy abans vista però és obvi que els nostres arguments poden fer-se amb altres mètriques fuzzy o amb mètriques clàssiques. La nostra decisió introdueix certs detalls de les mètriques (fuzzy) com alternativa de les distàncies clàssiques o mesures de similitud [64]. Per

altra part, els autors pensen que aquest tipus de mètrica fuzzy pot ser adequada en resolució de problemes d'enginyeria quan es fa servir la teoria fuzzy ja que les mesures són valors de dimensió en $[0, 1]$. Ara bé, aquesta mètrica M manca d'una propietat relevant des del punt de vista matemàtic: no és invariant a les translacions. Llavors la mesura resultant de M de dos parelles de vectors consecutius (o equidistants) no ha de ser necessàriament la mateixa. Augmentant el valor del paràmetre K anterior es redueix eixa *no uniformitat*. Aquest efecte es veu a la figura 4.1 on es comparen els valors donats per la mètrica fuzzy al calcular les distàncies entre el vector de color $[128, 128, 128]$ amb els vectors $[L, L, L]$, $L = 0, 1, \dots, 255$, així com el comportament del filtratge per a diferents valors de K . Vist això, el valor de K ha de prendre's suficientment alt per a que la *no uniformitat* siga reduïda, però no tan alt com per a evitar reduir la significativitat de les dades com pot deduir-se de (4.2).

4.2.1 Anàlisi Computacional de la Mètrica Fuzzy

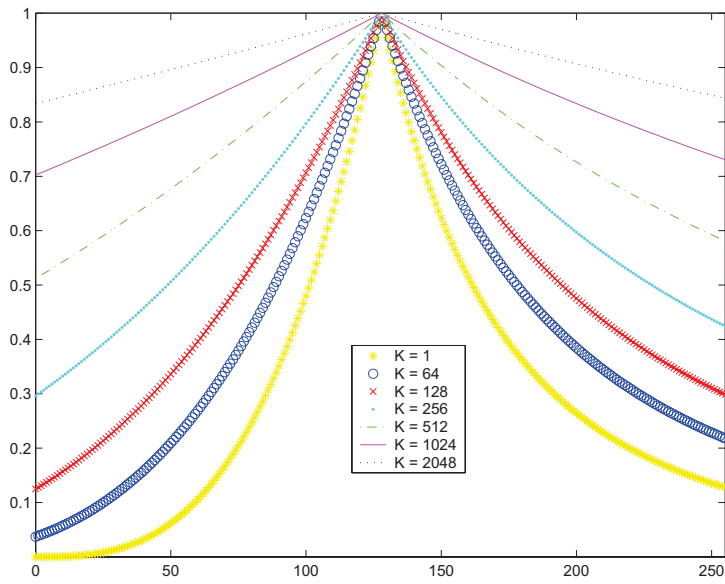
L'eficiència computacional de les distàncies és interessant en el camp dels filtres d'ordre estadístic [8, 9]. És per això que l'ús de la distància L_1 és de vegades més convenient que la distància L_2 [84]. El cas particular de la mètrica fuzzy proposada M és adequat per processament d'imatges de tres canals. Per a cada càlcul de M cal efectuar 3 comparacions, 6 sumes, 3 quocients i 2 productes. En el cas de la distància *city block* o L_1 són necessàries 3 comparacions (valor absolut), 3 restes i dos sumes mentre que per a la distància Euclídea o L_2 cal fer 3 restes, 3 potències, 2 sumes i una arrel quadrada. Com pot veure's a la taula 4.1 el cost computacional de M és major que el de L_2 i el de L_1 . Tot i això podem aplicar una optimització molt important al càlcul de M (representat com *fast - M*).

Taula 4.1. Comparativa computacional entre les mètriques clàssiques L_1 , L_2 i la mètrica proposada M mesurades en un Pentium IV 2.4GHz

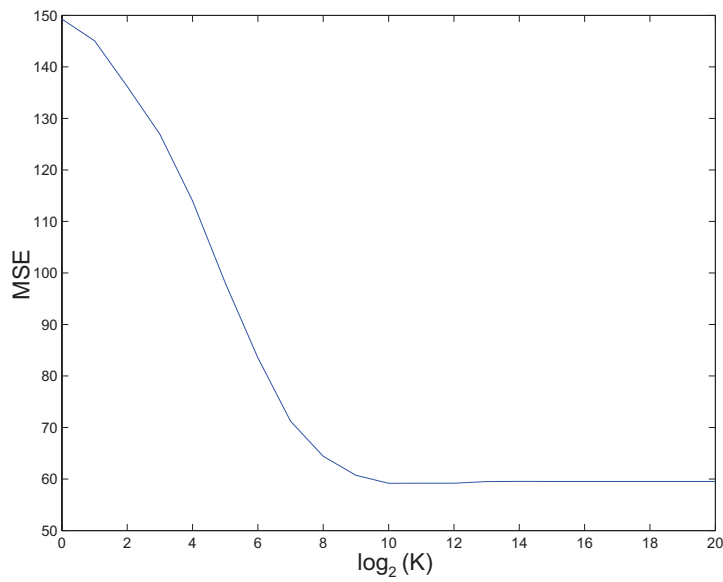
Mètrica	1 Càlculs (μs)	Càlculs per segon
L_1 Norm	28.37	$3.524 \cdot 10^4$
L_2 Norm	30.10	$3.322 \cdot 10^4$
M	34.68	$2.883 \cdot 10^4$
<i>Fast - M</i>	26.98	$3.706 \cdot 10^4$

Donat un valor fixe de K , el numerador i denominador de cada divisió en (4.1) està acotat entre $[K, 255 + K]$ al processar imatges a l'espai RGB. Per tant, totes les possibles divisions poden precalcular-se en una matriu quadrada C de 256×256 on

$$C(i, j) = \frac{\min\{i, j\} + K}{\max\{i, j\} + K} \quad i, j \in [0, 255] \quad (4.3)$$



(a)



(b)

Figura 4.1. Estudi del paràmetre K per a la mètrica M : (a) Valors obtinguts de M quan comparem el vector de color $[128, 128, 128]$ amb el vector de color $[L, L, L]$ on $L = 0, 1, \dots, 255$ per a diferents valors de K , (b) Valors obtinguts per a la mesura objectiva MSE (error quadrat mitjà) amb la imatge de prova Lenna per a diferents valors de K

Fent servir la matriu de precàlcul, el càlcul de $Fast - M$ per a dos píxels $I_i = (I_i(1), I_i(2), I_i(3))$, $I_j = (I_j(1), I_j(2), I_j(3))$ es redueix a

$$M_3(I_i, I_j) = \prod_{l=1}^3 C(I_i(l), I_j(l)) \quad (4.4)$$

Amb aquesta optimització, són suficients 3 accessos a la matriu i 2 productes per efectuar els càlculs.

El temps mesurat per construir la matriu de precàlcul C està al voltant de 0.8 segons a un Pentium IV 2.4GHz. A pesar que això suposa un cost inicial, el guany és aproximadament de $8\mu s$ (veure la Taula 4.1) en cada càlcul. Llavors el cost inicial es compensa quan s'han de calcular 10^5 (el que se sobrepassa amplement al filtratge d'una imatge de 50×50 píxels ¹).

Els resultats mostren que la mètrica fuzzy M és computacionalment més econòmica que les mètriques clàssiques L_1 i L_2 quan s'aplica l'optimització amb la matriu de precàlcul.

¹ Per a tots els filtres estudiats en aquest capítol, totes les imatges superen aquesta dimensió, per a finestres W amb 8 veïns de talla 3×3 .

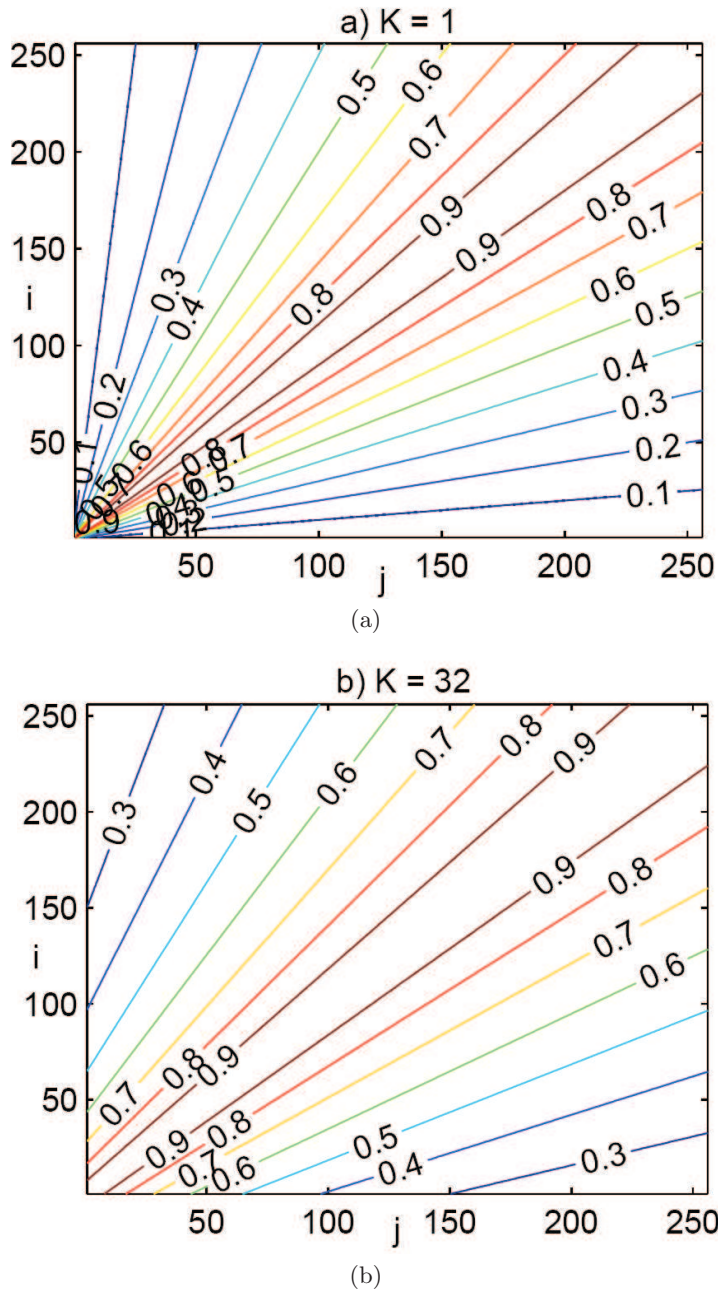


Figura 4.2. Contingut de la matriu de precàlcul $C(i, j)$ per a diferents valors de K : (a) $k = 1$, (b) $k = 32$

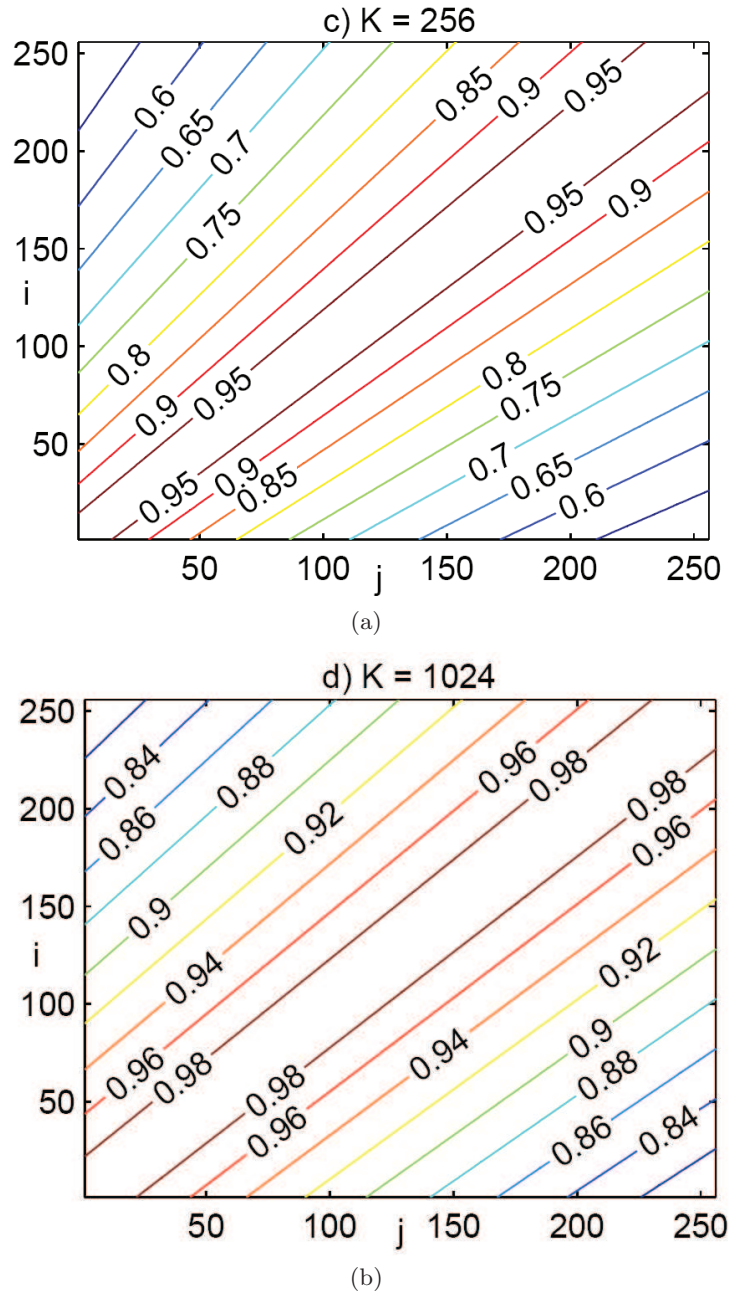


Figura 4.3. Contingut de la matriu de precàlcul $C(i, j)$ per a diferents valors de K : (a) $k = 256$, (b) $k = 1024$.

L'experimentació mostra [88] que els valors adequats de K per a construir la matriu de precàlcul i obtenir rendiments als filtratges adequats per a imatges a l'espai RGB estan entre [512, 2048], i a partir d'ara farem servir el valor de $K = 1024$ amb la mètrica *Fast* – M (representada simplement com a M). A la Figura 4.2 i la 4.2 pot veure's que el valor proposat de K manté pràcticament la simetria entre valors equidistants.

4.3 Detecció Proposada de Píxels Corruptes

El concepte de *peer group* introduït en [21, 49] s'ha utilitzat en diversos dissenys de filtratge d'imatges en color [21, 49, 75]. En aquest treball, fem una adaptació al context de les mètriques fuzzy del concepte de *peer group* de Smolka, [105]. En aquest concepte es crea un conjunt amb els píxels similars a un donat per a després decidir segons la cardinalitat del conjunt si el píxel és o no sorollós.

Una imatge en color a l'espai RGB normalment es representa com una matriu tridimensional $N_1 \times N_2 \times 3$, on cada píxel \mathbf{x}_i ($i \in N_1 \times N_2$) és un vector de tres components de valors enters prenent valors a l'interval $[0, 255]$. Llavors, d'acord al concepte de proximitat (recordeu l'axioma FM2 a la definició de mètrica fuzzy explicat a la Secció 4.2), per a un píxel central \mathbf{x}_i dins d'una finestra de filtratge W de grandària $n \times n$ i fixat $d \in]0, 1]$, representem per $\mathcal{P}(\mathbf{x}_i, d)$ el conjunt

$$\{\mathbf{x}_j \in W : M(\mathbf{x}_i, \mathbf{x}_j) \geq d\}$$

això vol dir que $\mathcal{P}(\mathbf{x}_i, d)$ és el conjunt de píxels de la finestra de filtratge W la distància fuzzy dels quals al píxel central \mathbf{x}_i no és menor que d . Òbviament, el conjunt $\mathcal{P}(\mathbf{x}_i, d)$ no és buit per a cada \mathbf{x}_i , ja que $M(x_i, x_i) = 1$.

Ara, fent servir la mateixa terminologia de [105], donat un número enter m , anomenarem com a *peer group* de m veïns (associat a \mathbf{x}_i) un subconjunt de $\mathcal{P}(\mathbf{x}_i, d)$ constituït per \mathbf{x}_i i $m (\leq n^2 - 1)$ píxels més. Concretament, si $\#\mathcal{P}(\mathbf{x}_i, d) = c + 1$ llavors $\mathcal{P}(\mathbf{x}_i, d)$ és un *peer group* de c veïns². Evidentment, per a cada $\mathcal{P}(\mathbf{x}_i, d)$ podem trobar un *peer group* de $m = 0$, però no ha d'existir necessàriament per a $m \geq 1$.

L'algorisme de detecció i eliminació de soroll impulsu, que anomenarem *filtre del peer group fuzzy modificat* (FMPGF), es basa en el concepte de *peer group modificat* aplicant-hi la mètrica fuzzy de la secció 4.2, i es descompon en dos passos: (1) detecció de soroll i (2) filtratge del soroll detectat al pas 1, separant el pas de la detecció en dos fases.

Pas 1 - Fase 1 (descriu a l'algorisme 1): la imatge a processar es divideix en $\frac{N_1 \cdot N_2}{n^2}$ finestres W' de dimensió $n \times n$, $n \in \{3, 5, 7, \dots\}$. Sent \mathbf{x}_i el píxel central de cadascuna de les finestres i donat un paràmetre positiu d , es computa el seu *peer group* $\mathcal{P}(\mathbf{x}_i, d)$. Una vegada establert el valor m (habitualment per

² $\#$ representa la cardinalitat del conjunt.

Algorisme 1: FMPGF - Pas 1 Detecció de Píxels Sorollosos. Fase 1Entrades : $m, d, n, \text{Imatge_Original}$ Sortides : Matriu_Soroll

```

1 Obtenir  $N_1, N_2$ , dimensions de  $\text{Imatge\_Original}$ ;
2 Construir  $N_1 * N_2 / n^2$  finestres disjunctes ( $W'$ );
3 PerATot  $W'$  Fer
4   | Construir  $\mathcal{P}(\mathbf{x}_i, d)$  en  $W'$ ;
5   | Si ( $\#\mathcal{P}(\mathbf{x}_i, d) == m + 1$ ) Llavors
6   |   |  $\forall \mathbf{x}_j \in \mathcal{P}(\mathbf{x}_i, d)$ ,  $\mathbf{x}_j$  es declara com no corrupte ;
7   |   |  $\forall \mathbf{x}_k \in W', \mathbf{x}_k \notin \mathcal{P}(\mathbf{x}_i, d)$ ,  $\mathbf{x}_k$  es declara com no diagnosticat ;
8   | Si no
9   |   |  $\mathbf{x}_i$  es declara com corrupte provisionalment ;
10  |   |  $\forall \mathbf{x}_j \in W', j \neq i$ ,  $\mathbf{x}_j$  es declare com no diagnosticat ;
11  | FiSi
12  | Transcriure  $W'$  a  $\text{Matriu\_Soroll}$ ;
13 FiPer

```

finestres $n \times n, m = n - 1$, així si $n = 3, m = 2$) si el *peer group* conté almenys $m + 1$ elements, aleshores el píxel central (x_i) es marca com *no corrupte* i els membres del *peer group* com *no corruptes*. En cas contrari el central es declara com a *corrupte* deixant els membres del *peer group* com *no diagnosticats*, a l'igual com els membres de la finestra que no pertanyen al *peer group*. Al final d'aquest pas, la matriu de soroll és una matriu amb valors de 0, 1 i 2 amb la següent estructura:

$$\text{Matriu_Soroll}_{i,j} = \begin{cases} 0 & \text{El píxel } (i, j) \text{ de la imatge és } \textit{no corrupte} . \\ 1 & \text{El píxel } (i, j) \text{ de la imatge és } \textit{corrupte} . \\ 2 & \text{El píxel } (i, j) \text{ de la imatge és } \textit{no diagnosticat} . \end{cases}$$

Algorisme 2: FMPGF - Pas 1 Detecció de Píxels Sorollosos. Fase 2Entrades : $m, d, n, \text{Imatge_Original}, \text{Matriu_Soroll}$ Sortides : Matriu_Soroll

```

1 Seleccionar de  $\text{Matriu\_Soroll}$  totes els  $\mathbf{x}_i$  no diagnosticats;
2 PerATot  $W'$  finestra ( $n \times n$ ) amb  $\mathbf{x}_i$  com a píxel central Fer
3   | Construir  $\mathcal{P}(\mathbf{x}_i, d)$  en  $W'$ ;
4   | Si ( $\#\mathcal{P}(\mathbf{x}_i, d) == m + 1$ ) Llavors
5   |   |  $\forall \mathbf{x}_j \in \mathcal{P}(\mathbf{x}_i, d)$ ,  $\mathbf{x}_j$  es declara com no corrupte ;
6   | Si no
7   |   |  $\mathbf{x}_i$  es declara com corrupte ;
8   | FiSi
9   | Transcriure  $W'$  a  $\text{Matriu\_Soroll}$ ;
10 FiPer

```

Pas 1 - Fase 2 (descriu a l'algorisme 2): cada píxel *no diagnosticat* \mathbf{x}_i és el central d'una finestra $n \times n$. Si aconseguim trobar un *peer group* de cardinalitat $m + 1$, tant el central com els membres del *peer group* es declaren *no corruptes*, declarant-lo *corrupte* en cas contrari. Al final d'aquest pas, la matriu de soroll és una matriu de zeros i uns amb la següent estructura:

$$\text{Matriu_Soroll}_{i,j} = \begin{cases} 0 & \text{El píxel } (i, j) \text{ de la imatge és } \textit{no corrupte} . \\ 1 & \text{El píxel } (i, j) \text{ de la imatge és } \textit{corrupte} . \end{cases}$$

Pas 2: s'aplica una tècnica de filtratge per substitució per a calcular la sortida \mathbf{y} de les entrades \mathbf{x} dins una finestra W de talla $n \times n$ com es veu a l'algorisme 3. Per a computar la sortida si el píxel central de W és *no corrupte* s'aplica l'operació identitat (no es canvia), mentre que si és *corrupte* es calcula el AMF sobre els píxels *no corruptes* de la finestra W . Pot ocórrer el cas que en W tots foren *corruptes*, amb la qual cosa seria necessari augmentar la grandària de W fins a trobar almenys un píxel del veïnat del píxel central de W *no corrupte*.

Algorisme 3: FMPGF - Pas 2 Aplicació de la Tècnica de Filtratge per Substitució

Entrades : n , imatge_Original , Matriu_Soroll
Sortides : imatge_Filtrada

```

1 Obtenir  $N_1, N_2$ , dimensions de  $\text{imatge\_Original}$ ;
2 Per A  $i = 1$  fins  $N_1$  Fer
3   Per A  $j = 1$  fins  $N_2$  Fer
4     Si ( $\text{Matriu\_Soroll}_{(i,j)} == 1$ ) Llavors
5        $W =$  Finestra amb centre  $\mathbf{x}_{(i,j)}$  de talla  $n \times n$ ;
6       Mentre ( $\#(\text{píxels no corruptes de } W) == 0$ ) Fer
7          $n = n + 1$ ;
8          $W =$  Finestra amb centre  $\mathbf{x}_{(i,j)}$  de talla  $n \times n$ ;
9       FiMentre
10       $\text{imatge\_Filtrada}_{(i,j)} = \text{AMF}_{out}$  sobre els píxels no corruptes de  $W$ ;
11     Si no
12        $\text{imatge\_Filtrada}_{(i,j)} = \text{imatge\_Original}_{(i,j)}$ ;
13     FiSi
14   FiPer
15 FiPer

```

Pot veure's un esquema visual del funcionament del filtre a les Figures 4.4, 4.5, 4.6 i 4.7, on es veuen les imatges originals, contaminades, la matriu de soroll (on els punts negres indiquen que el píxel corresponent és sorollós a la imatge contaminada) produïda després de la fase 1 i finalment la imatge resultant d'aplicar el filtre.

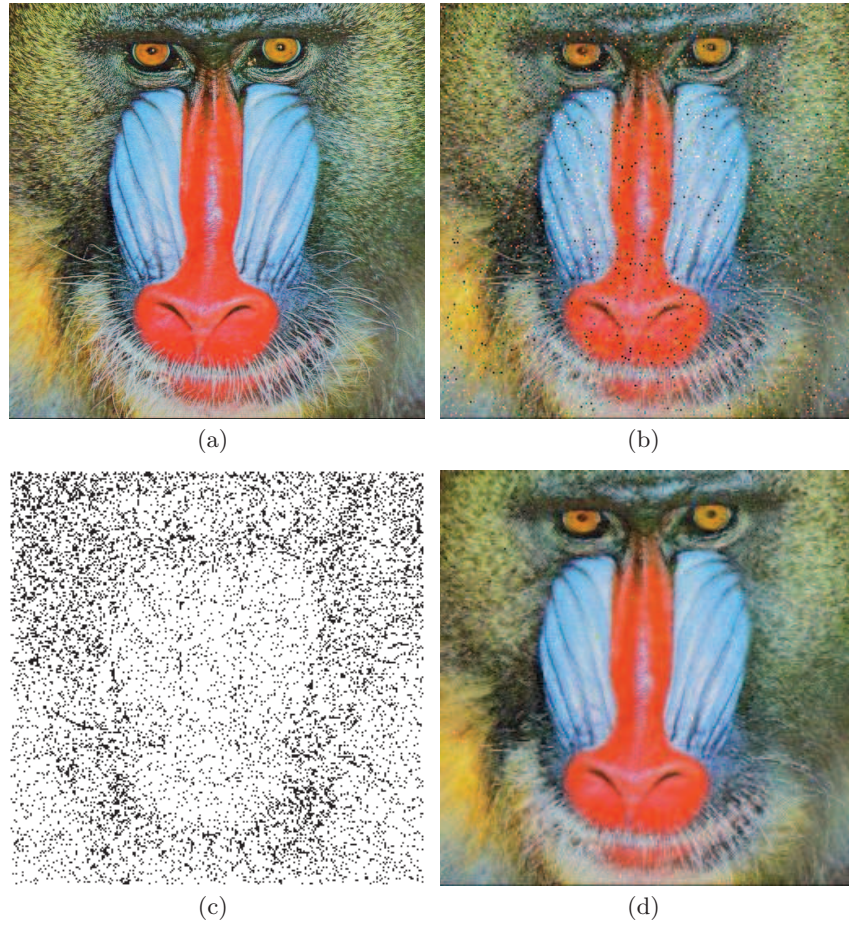


Figura 4.4. Guia visual del procés de filtratge: (a) Imatge Baboon original, (b) imatge Baboon contaminada amb un 10% de soroll impulsiu, (c) soroll detectat a la imatge, (d) sortida del filtre FMPGF

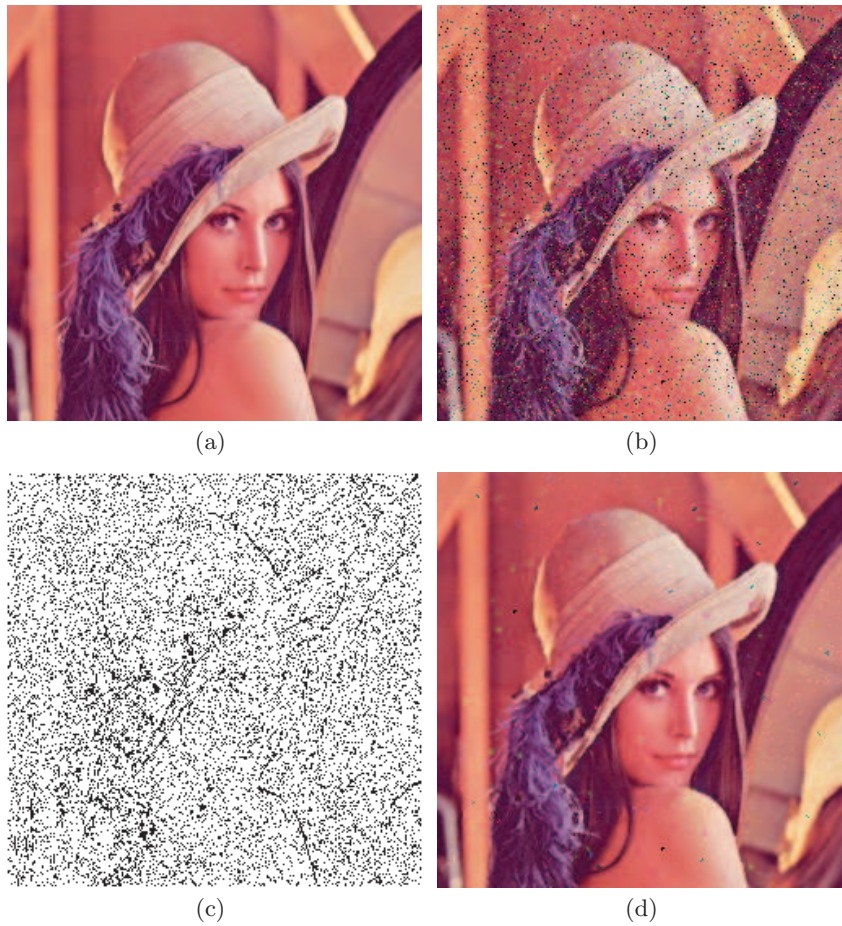


Figura 4.5. Guia visual del procés de filtratge: (a) Imatge Lenna original, (b) imatge Lenna contaminada amb un 20% de soroll impulsiu, (c) soroll detectat a la imatge, (d) sortida del filtre FMPGF

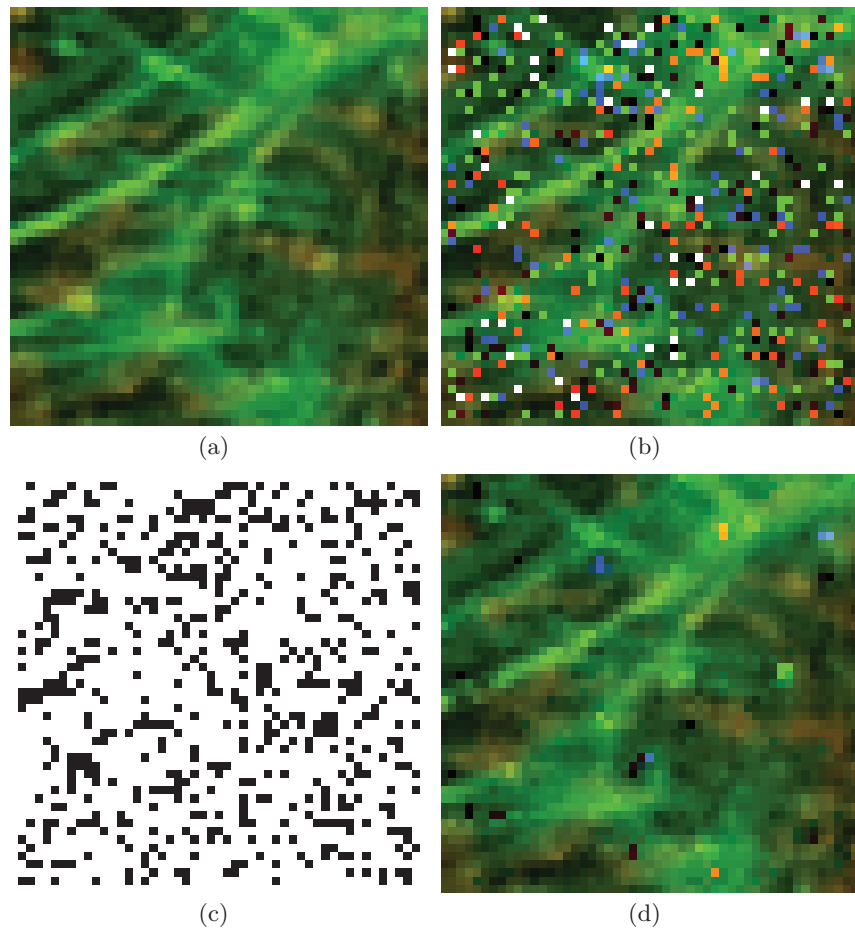


Figura 4.6. Guia visual del procés de filtratge: (a) Imatge Micro original, (b) imatge Micro contaminada amb un 30% de soroll impulsiu, (c) soroll detectat a la imatge, (d) sortida del filtre FMPGF

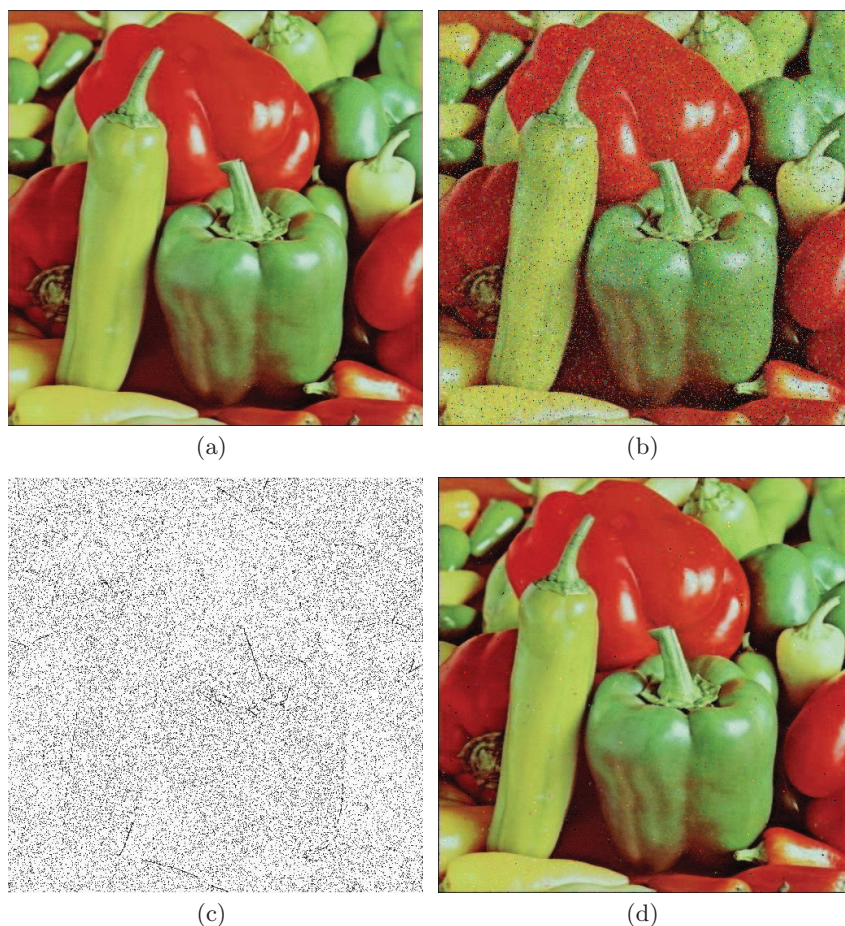


Figura 4.7. Guia visual del procés de filtratge: (a) Imatge Peppers original, (b) imatge Peppers contaminada amb un 15% de soroll impulsiu, (c) soroll detectat a la imatge, (d) sortida del filtre FMPGF

La principal idea subjacent de l'esquema que presentem és que quan es troba un *peer group* $\mathcal{P}(\mathbf{x}_i, d)$, $\#\mathcal{P}(\mathbf{x}_i, d) = m + 1$ per al píxel central \mathbf{x}_i , tots els $m + 1$ píxels del *peer group* es declaren com no corruptes. Això és un comportament intuïtiu ja que si un píxel és *no corrupte*, aquells píxels que són propers a ell (les seues “parelles” dins del *peer group*) són probablement *no corruptes*. Noteu que aplicant l'axioma (FM4) de la definició de la mètrica fuzzy a la Secció 4.2, pot garantir-se que la distància fuzzy entre dos parells de píxels del *peer group* serà major o igual que d^2 . Llavors, seleccionant un valor adequat de d , aquells píxels que formen part del *peer group* poden declarar-se com *no corruptes* amb suficient confiança, el que suposa la idea intuïtiva

abans mencionada. Aquesta operació redueix de manera dràstica el nombre de càlculs a efectuar (i es prova formalment a la Secció 4.5) sense minvar significativament el rendiment del procés de filtratge, com podrà veure's a la Secció 4.4. L'ús de finestres disjunes a la fase 1 del pas 1 s'ha fet per accelerar la velocitat de l'algorisme. Atès que després de la primera fase hi haurà píxels declarats com *no diagnosticats*, la fase 2 és necessària per a completar els diagnòstics i pot ocórrer que píxels que en la primera fase s'hagen declarat com *corruptes* provisionalment passen a ser *no corruptes*. Al pas 2 s'efectua l'operació de substitució triant entre l'operació identitat amb la imatge original o la sortida del AMF (imitant [105]). S'aplica el filtre AMF per la seua simplicitat i baix cost computacional, encara que altres tècniques de filtratge adequades com el VMF o similars s'hagueren pogut aplicar en les mateixes condicions sense perdre generalitat.

4.4 Estudi Experimental i Comparativa de Rendiment

A aquesta secció, s'han considerat distintes imatges per avaluar el rendiment del filtre proposat FMPGF explicat a la Secció 4.3. S'ha fet servir el model de soroll impulsiu per a la transmissió de soroll, tal i com es defineix a [84], per contaminar les imatges de Baboon, Lenna, Micro i Peppers (veure Figura 4.8).

Taula 4.2. Filtres presos en consideració per a la comparació del rendiment juntament amb la notació

Notació	Filtre
AMF	Filtre de Mitjana Aritmètica[84]
VMF	Filtre de Mediana Vectorial[4]
DDF	Filtre de Distància Direccional [47]
BVDF	Filtre Bàsic de Distància Vectorial [113]
FIVF	Filtre Ràpid Vectorial Impulsiu[88]
PGF	Filtre de Peer Group [105]
FMPGF	Filtre de Peer Group Fuzzy Modificat

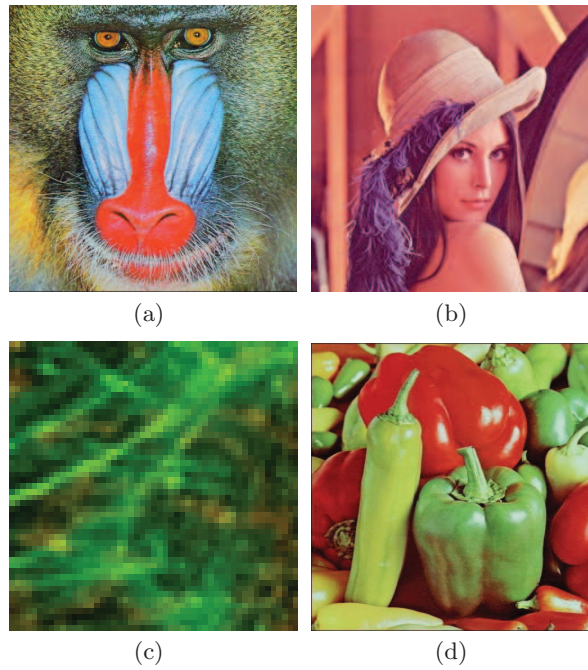


Figura 4.8. Imatges de test: (a) Baboon, (b) Lenna, (c) Micro, (d) Peppers.

El filtre proposat s'ha comparat front als filtres recents com els clàssics amb bona habilitat de preservació de detalls, i s'enumeren a la Taula 4.2, fent servir com a mesures de qualitat objectives l'*Error Absolut Mitjà* (MAE), el *Ratio de Pics de Soroll a la Senyal* (PSNR) i la *Diferència de Color Normalitzada* (NCD), definits al Capítol 2.

Abans d'abordar el rendiment del filtre FMPGF davant d'altres filtres vectorials, s'ha analitzat el comportament del filtre per ajustar els paràmetres m i d . Els resultats poden veure's a la Figures 4.9 i 4.10, i com pot deduir-se d'ella els valors més adequats han estat per a $m = 2, 3$ i $d \in [0.93, 0.97]$.

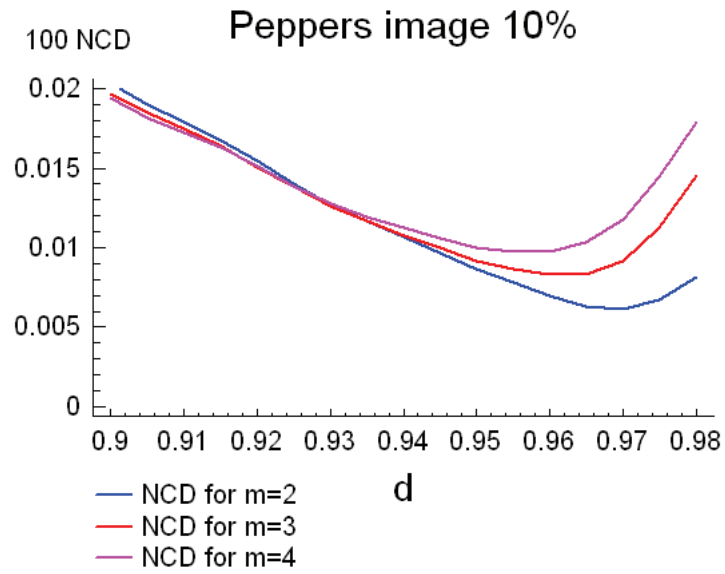


Figura 4.9. Ajust de m i d per a la imatge 'Peppers' amb una contaminació del 10% de soroll impulsiu

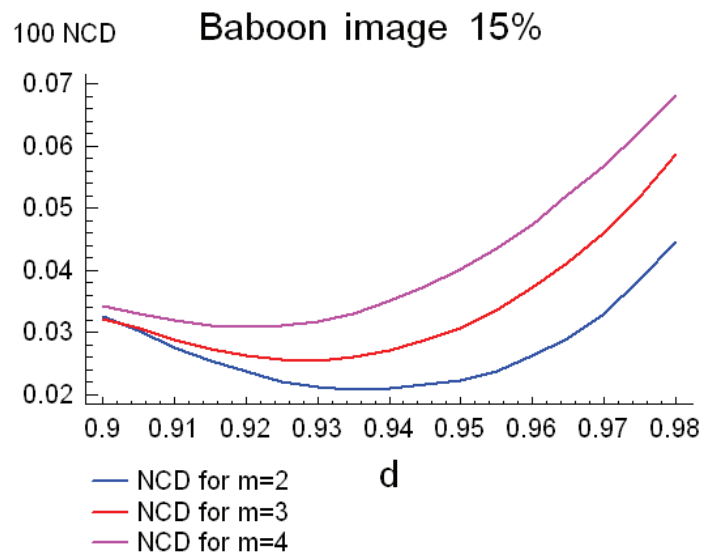


Figura 4.10. Ajust de m i d per a la imatge 'Baboon' amb una contaminació del 15% de soroll impulsiu

S'han triat per ajustar els paràmetres una imatge amb molta textura i poc de detall (Peppers) i altra amb molt de detall i poca textura (Baboon). Com pot observar-se a la figura 4.9 amb un valor inicial de $d = 0.9$ s'obtenen valors alts del NCD. Aquests valors van disminuint segons augmenta d fins trobar un valor òptim de $d = 0.97$. Com que podem variar el paràmetre m , s'han fet proves per a $m = 2, 3, 4$ i com s'ha exposat anteriorment el valor òptim és $m = 2$ (recordeu que el valor adequat de $m = n - 1$, sent n la grandària de la finestra d'anàlisi). A la Figura 4.10 s'observa l'estudi similar per a la imatge Baboon, en el qual el valor òptim el trobem al voltant de $d = 0.94$, mantenint també el paràmetre $m = 2$.

Després d'això, s'han efectuat distintes experimentacions en tal de confrontar el filtre FMPGF davant del filtre rival més competitiu PGF [105]. S'han pres mesures de rendiment en termes de qualitat objectiva (NCD) i els seus temps computacionals relatius al VMF³, comparant-se en distints nivells de contaminació i per a distints valors de m . Poden observar-se els resultats de l'estudi a les figures 4.11 a 4.14. El temps computacional es mostra com a percentatge respecte del filtre VMF per a mostrar més clàrament la millora aconseguida. Pot veure's que el rendiment en termes de mesures de qualitat es bastant similar mentre que el temps computacional és sempre a favor del filtre proposat FMPGF.

En les figures 4.11 i 4.12 es mostra l'estudi comparatiu de la imatge Baboon, imatge amb un alt nivell de detall i poques zones amb textures homogènies. En aquesta imatge d'observa que el funcionament del nostre algorisme és millor en nivells de soroll baixos, obtenint-se resultats similars al voltant del 7% de soroll, moment a partir del qual es comença a perdre qualitat respecte del PGF. Per contra la millora computacional sempre està al voltant de 2 punts millor que el PGF, cosa que es manté estable a mesura que augmenta la intensitat de soroll. En la mateixa imatge però fent servir el paràmetre $m = 3$ s'observa que el rendiment computacional és millor (4 punts) ja que el nostre algorisme és capaç de diagnosticar més d'un píxel com *no corrupte* i el PGF amb $m = 3$ és menys eficaç ja que exigint més diagnostica el mateix. S'observa que el rendiment és millor que el PGF fins arribar a intensitats altes de soroll (pràcticament el 20%), però amb valors numèrics pitjors que amb $m = 2$.

En les figures 4.13 i 4.14 es mostra l'estudi comparatiu de la imatge Peppers, imatge amb un baix nivell de detall i moltes zones amb textures homogènies, al contrari de Baboon. En aquest cas el cost computacional s'estreta més que en Baboon però per contra podem veure que començant amb un rendiment de filtratge similar, conforme s'augmenta la intensitat de soroll el FMPGF va obtenint un millor rendiment. Per a $m = 3$ s'observa a l'igual que l'anterior imatge un millor cost computacional que el PGF davant $m = 2$ però en quant al rendiment del filtratge s'observa un resultat molt similar al PGF.

³ Resultats obtinguts en un Pentium IV, 3.4 GHz, 512 MB RAM.

Com a estudi adicional, a la Taula 4.3 es mostren les taxes d'encerts i errades, tant positives com negatives, dels dos filtrats. Aquestes variables estudiades responen a:

Encerts Positius	Píxels corruptes detectats com corruptes
Encerts Negatius	Píxels no corruptes detectats com no corruptes
Errades Positives	Píxels corruptes detectats com no corruptes
Errades Negatives	Píxels no corruptes detectats com corruptes

Taula 4.3. Ratio d'encerts/errades per als filtres FMPGF i PGF.

Imatge	Filtre	Encerts	Errades	Encerts	Errades
		Negatius	Negatives	Positius	Positives
Baboon original	FMPGF	0.994	0.006	0.000	0.000
	PGF	0.963	0.037	0.000	0.000
Baboon 5%	FMPGF	0.934	0.017	0.0393	0.010
	PGF	0.910	0.041	0.046	0.003
Baboon 10%	FMPGF	0.874	0.030	0.082	0.013
	PGF	0.859	0.046	0.090	0.005
Baboon 15%	FMPGF	0.813	0.041	0.128	0.018
	PGF	0.805	0.050	0.137	0.008
Baboon 20%	FMPGF	0.746	0.058	0.174	0.022
	PGF	0.747	0.057	0.183	0.013
Peppers original	FMPGF	0.999	0.001	0.000	0.000
	PGF	0.998	0.002	0.000	0.000
Peppers 5%	FMPGF	0.939	0.011	0.046	0.004
	PGF	0.948	0.002	0.044	0.006
Peppers 10%	FMPGF	0.882	0.018	0.094	0.006
	PGF	0.898	0.003	0.087	0.012
Peppers 15%	FMPGF	0.833	0.019	0.138	0.010
	PGF	0.849	0.003	0.129	0.019
Peppers 20%	FMPGF	0.788	0.015	0.178	0.019
	PGF	0.799	0.004	0.170	0.027

Com pot veure's, el filtre FMPGF té una millor habilitat en la preservació dels detalls per a imatges amb textures (el cas de 'Baboon'), encara que en aquestes imatges certes vegades es falla en la detecció de baixos impulsos (casos on el soroll és proper al fons on estan). En altres tipus d'imatges amb amples zones homogènies (com 'Peppers'), el filtre FMPGF és més adequat que el filtre PGF en la detecció del soroll però la capacitat de preservar els detalls és menor. D'aquests resultats podem concloure que el filtre que proposem presenta un rendiment competitiu respecte al filtre PGF mentre

que millora el cost computacional (com es prova a la Secció 4.5). Tenir en compte que el filtre PGF està introduït a [105] com un esquema adequat per a aplicacions en temps real i és un dels més ràpids en l'estat de l'art.

Les Taules 4.4-4.7 mostren una comparativa més general del filtre FMPGF davant dels filtres vectorials de la Taula 4.2 fent servir unes bateries d'imatges contaminades amb distintes densitats de soroll impulsiu. Pot veure's que el filtre proposat presenta un rendiment competitiu respecte als filtres recents de l'estat de l'art, millorant significativament el rendiment dels filtres clàssics. Algunes de les imatges de sortida dels filtres poden veure's a les Figures 4.15, 4.16 per a una comparativa visual.

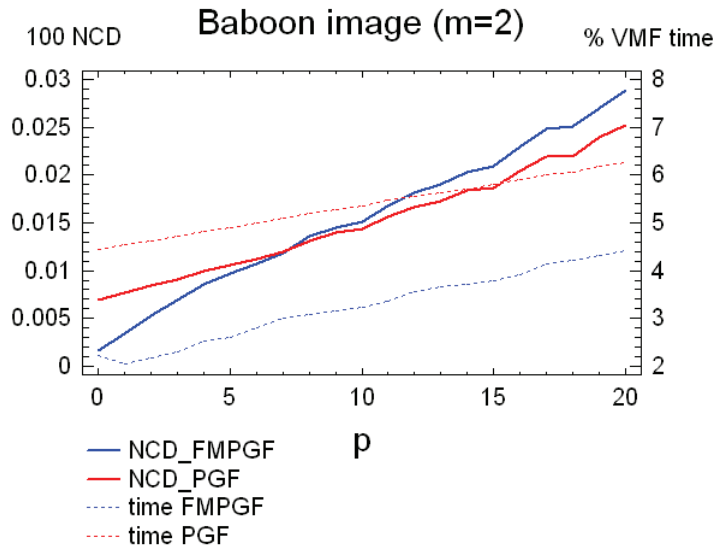


Figura 4.11. Comparativa FMPGF vs. PGF en la imatge Baboon per a $m = 2$

4.5 Complexitat Computacional

En aquesta secció, amb un argument similar a l'emprat a [105], compararem la càrrega computacional del filtre proposat amb el filtre PGF i el filtre VMF. En primer lloc, cal tenir en compte que una part de la millora computacional s'obté pel fet de la substitució de les mètriques clàssiques per la mètrica fuzzy proposada, ja que és computacionalment més eficient que les mètriques clàssiques [88]. En tot cas, l'operació amb cost computacional més demandada és el càlcul de distàncies entre els píxels de les imatges en color i és per això que compararem el nombre total de càlculs de distàncies necessàries en aquest

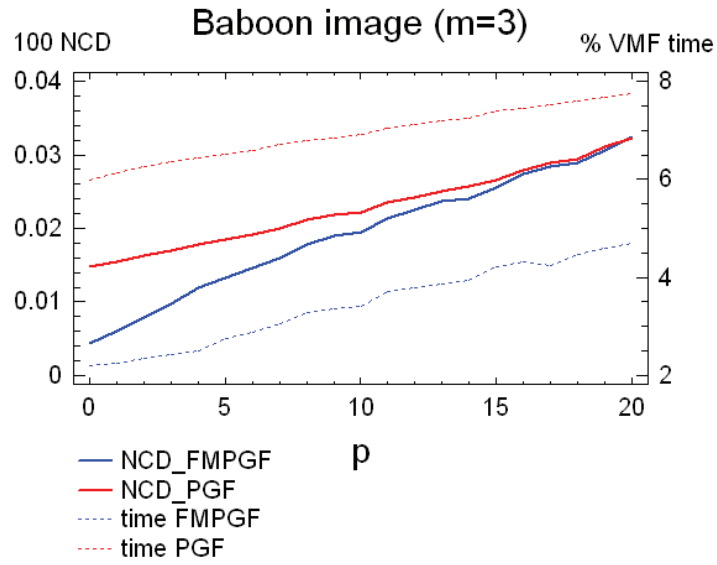


Figura 4.12. Comparativa FMPGF vs. PGF en la imatge Baboon per a $m = 3$

Taula 4.4. Comparativa de rendiment per a la imatge Baboon (256×256).

Filtre	5%			10%			15%		
	MAE	PSNR	NCD (10^{-2})	MAE	PSNR	NCD (10^{-2})	MAE	PSNR	NCD (10^{-2})
AMF	19.812	20.338	8.140	20.465	19.997	9.386	20.933	19.734	10.514
VMF	10.878	23.116	4.496	10.962	23.073	4.541	11.083	22.999	4.607
DDF	10.708	23.029	4.201	10.920	22.952	4.311	11.164	22.838	4.442
BVDF	11.891	22.065	4.548	12.147	21.930	4.661	12.585	21.723	4.853
FIVF	1.698	27.055	1.555	2.891	25.519	2.258	4.012	24.440	2.963
PGF $_{m=2}$	2.678	26.931	1.062	3.543	26.007	1.439	4.386	25.275	1.867
FMPGF $_{m=2}$	2.163	27.738	1.007	3.047	26.508	1.519	4.209	25.274	2.089
PGF $_{m=3}$	4.874	24.483	1.857	5.714	23.962	2.220	6.570	23.486	2.661
FMPGF $_{m=3}$	3.740	25.373	1.551	4.634	24.667	2.035	5.526	24.079	2.548

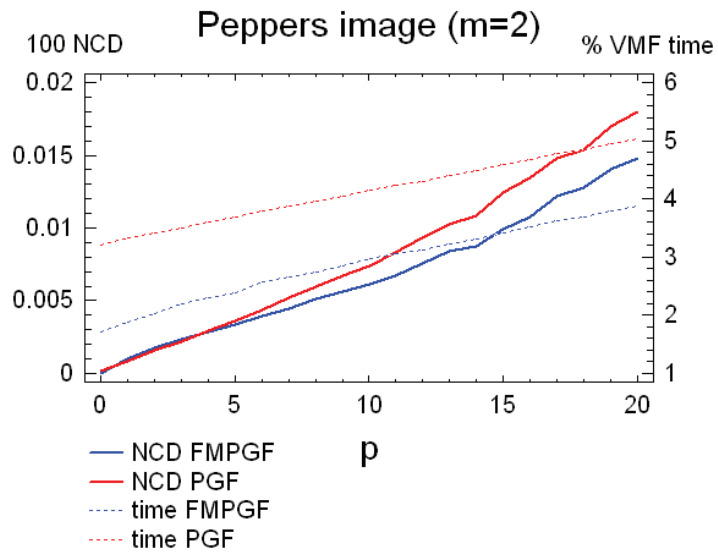


Figura 4.13. Comparativa FMPGF vs. PGF en la imatge Peppers per a $m = 2$

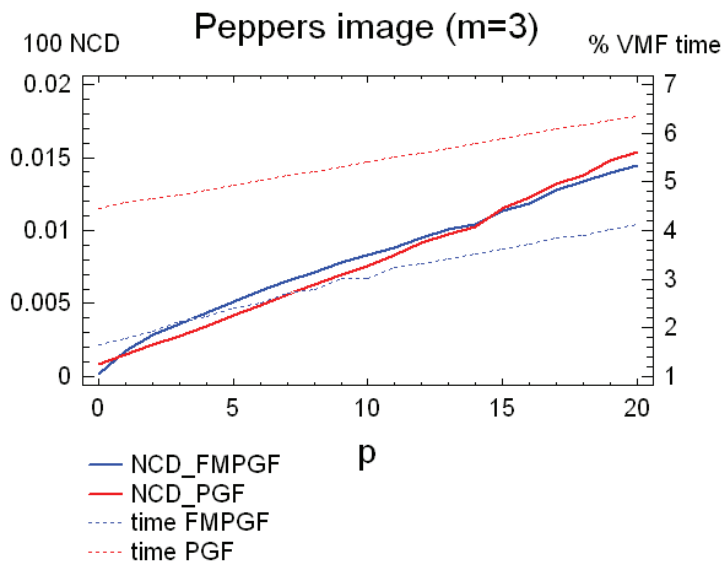


Figura 4.14. Comparativa FMPGF vs. PGF en la imatge Peppers per a $m = 2$

Taula 4.5. Comparativa de rendiment per a la imatge Lenna (256×256).

Filtre	5%			10%			15%		
	MAE	PSNR	NCD (10^{-2})	MAE	PSNR	NCD (10^{-2})	MAE	PSNR	NCD (10^{-2})
AMF	16.415	22.422	7.358	16.825	21.947	8.798	17.383	21.483	10.260
VMF	2.755	32.051	1.781	2.874	31.767	1.847	2.946	31.611	1.903
DDF	2.764	31.808	1.663	2.982	31.325	1.788	3.159	31.006	1.915
BVDF	2.985	31.382	1.720	3.248	30.773	1.869	3.513	30.112	2.035
FIVF	0.414	35.685	0.482	0.743	34.158	0.669	1.031	32.972	0.936
PGF _{m=2}	0.404	37.886	0.297	0.696	35.257	0.553	1.081	32.397	0.915
FMPGF _{m=2}	0.521	36.196	0.370	0.900	33.390	0.687	1.354	30.897	1.077
PGF _{m=3}	0.7386	34.384	0.445	1.072	32.857	0.705	1.442	31.639	0.982
FMPGF _{m=3}	0.690	34.456	0.529	1.240	31.805	0.868	1.677	30.572	1.173

Taula 4.6. Comparativa de rendiment per a la imatge Micro (256×256).

Filtre	5%			10%			15%		
	MAE	PSNR	NCD (10^{-2})	MAE	PSNR	NCD (10^{-2})	MAE	PSNR	NCD (10^{-2})
AMF	11.558	23.406	12.152	12.819	22.500	14.389	15.312	21.074	16.150
VMF	4.805	29.276	4.371	4.937	29.014	4.541	5.103	28.849	4.641
DDF	4.869	29.159	4.188	5.070	28.809	4.445	5.470	28.393	4.671
BVDF	6.206	27.091	5.104	6.623	26.154	5.504	6.808	26.345	5.563
FIVF	0.839	34.314	0.976	1.385	31.815	1.795	1.735	31.261	2.022
PGF _{m=2}	0.688	36.582	0.790	1.221	31.755	1.752	1.810	28.834	2.239
FMPGF _{m=2}	0.845	35.296	0.862	1.666	29.606	1.949	2.167	27.391	2.554
PGF _{m=3}	1.009	34.123	0.962	1.475	32.281	1.522	2.085	30.306	2.019
FMPGF _{m=3}	0.817	35.324	0.981	1.701	31.095	1.869	2.131	30.449	2.141

Taula 4.7. Comparativa de rendiment per a la imatge Peppers (512×512).

Filtre	5%			10%			15%		
	MAE	PSNR	NCD (10^{-2})	MAE	PSNR	NCD (10^{-2})	MAE	PSNR	NCD (10^{-2})
AMF	15.370	22.412	9.904	16.260	21.767	11.776	17.200	21.211	13.390
VMF	2.171	35.167	1.915	2.252	34.771	2.000	2.337	34.251	2.088
DDF	2.196	34.973	1.802	2.358	34.387	1.940	2.515	33.884	2.069
BVDF	2.525	32.771	1.957	2.744	31.696	2.123	2.994	30.519	2.300
FIVF	0.537	35.703	0.876	0.746	34.687	1.207	0.974	33.591	1.517
PGF _{m=2}	0.513	35.823	0.739	0.828	32.761	1.241	1.192	30.622	1.802
FMPGF _{m=2}	0.690	34.406	0.610	1.038	31.522	0.991	1.424	29.528	1.476
PGF _{m=3}	0.669	33.533	0.758	0.924	32.616	1.149	1.214	31.347	1.536
FMPGF _{m=3}	0.987	31.948	0.822	1.362	30.681	1.121	1.735	29.425	1.440

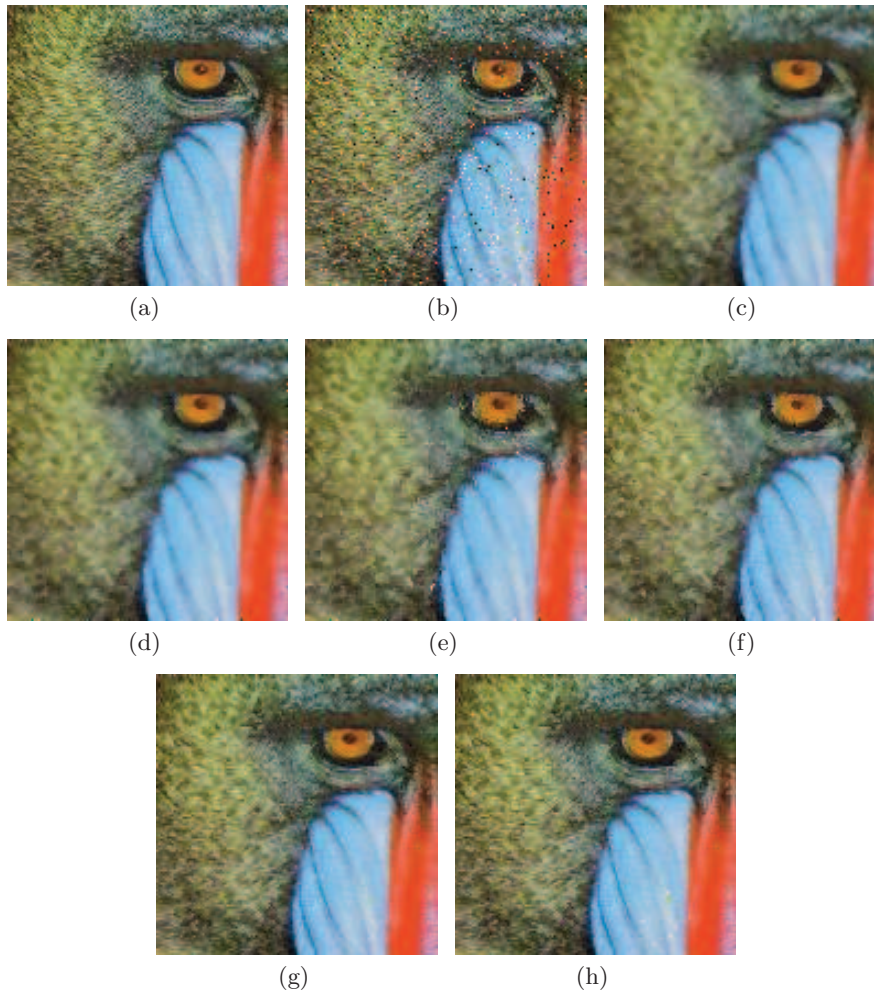


Figura 4.15. Images de sortida dels filtres estudiats: (a) Detall de la imatge Baboon, (b) detall de la imatge Baboon contaminada amb un 10% de soroll impulsiu, (c) sortida del filtre VMF, (d) sortida del filtre DDF, (e) sortida del filtre BVFD, (f) sortida del filtre FIVF, (g) sortida del filtre PGF, (h) sortida del filtre FMPGF.

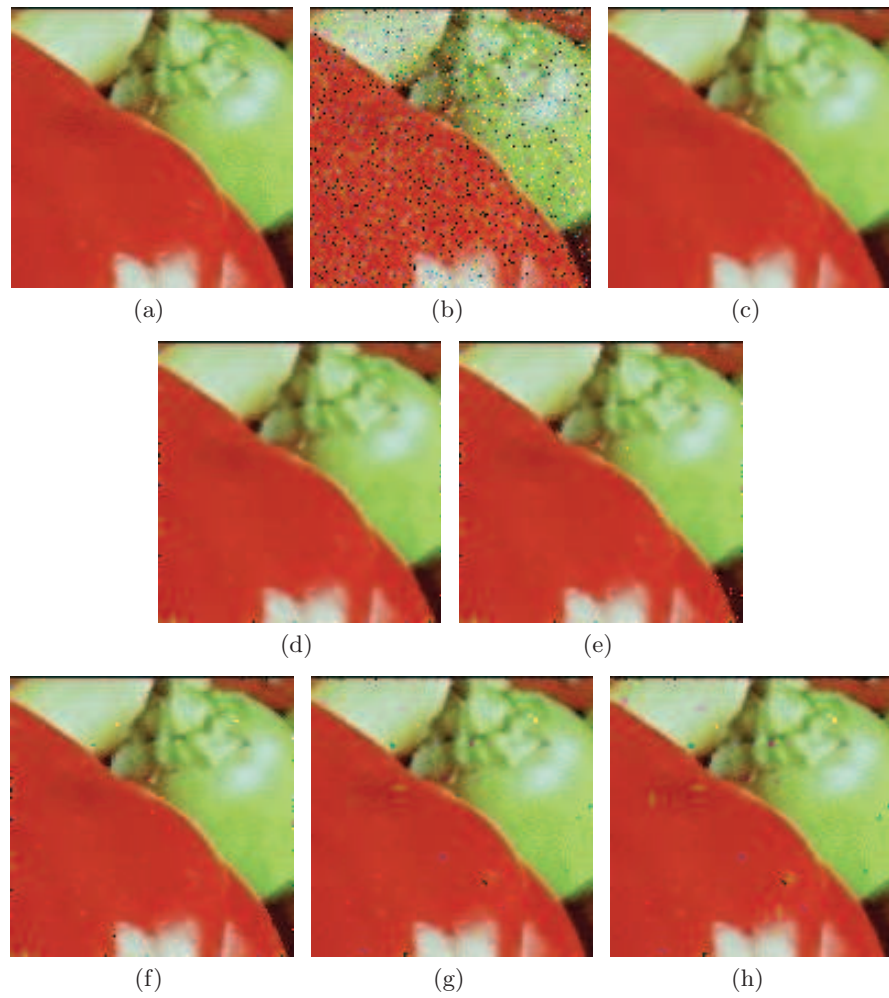


Figura 4.16. Images de sortida dels filtres estudiats: (a) Detall de la imatge Peppers, (b) detall de la imatge Peppers contaminada amb un 20% de soroll impulsiu, (c) sortida del filtre VMF, (d) sortida del filtre DDF, (e) sortida del filtre BVFD, (f) sortida del filtre FIVE, (g) sortida del filtre PGF, (h) sortida del filtre FMPGF.

algorisme. El gran guany del nostre mètode és el fet que quan un píxel central es diagnostica com a no corrupte llavors som capaços de diagnosticar altres píxels com a no corruptes.

Seguint la notació de [105], siga η el nombre de veïns del píxel central en la finestra de filtratge W (per a finestres $n \times n$, $\eta = n^2 - 1$). Per trobar quantes distàncies es calculen de mitjana per al diagnòstic d'un píxel dins de W , definirem a continuació les següents variables:

Siguen r i q les probabilitats que un píxel \mathbf{x}_j del veïnat del píxel central \mathbf{x}_i en W siga membre del *peer group* de m elements, quan el píxel central siga no corrupte o corrupte, respectivament. Les probabilitats r i q depenen de l'estructura de la imatge, estadístiques i intensitats del soroll, així com del paràmetre de la distància fuzzy d , i podem suposar que $r < 1 - p$ i $q < p$. Com que l'expressió (8) dels autors [105] resulta confusa en la seua notació i no fa cap argumentació, ni ens deixa clar sota quines condicions s'atura l'algorisme, nosaltres farem un argument que clarifiqui aquests aspectes i donarem una expressió que ens serà útil per entendre-ho.

Posem per cas una finestra 5×5 , amb la qual cosa el píxel central tindrà $\eta = 24$ veïns. Ordenem els 24 veïns com $k = 1, 2, \dots, 24$. Es programa l'algorisme per a que quan trobe $m = 4$ (recordeu que $m = n - 1$) veïns propers pugui declarar-se el central com *no corrupte*. L'algorisme realitzarà la recerca per la finestra fins el píxel 24 si no troba 4 veïns propers. Ací tenim una primera millora a l'algorisme, que és aturar la recerca en cas que no pugui arribar a trobar m píxels. Si entre els 24 hi han menys de 4 propers, declararà el central com *corrupte*.

Les expressions (8) i (9) de Smolka [105]

$$f_1(k) = \begin{cases} \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} r & \text{si } k < \eta \\ \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} r + \sum_{l=0}^{m-1} \binom{\eta}{l} r^l (1-r)^{\eta-l} & \text{si } k = \eta \end{cases} \quad (4.5)$$

$$f_2(k) = \begin{cases} \binom{k-1}{m-1} q^{m-1} (1-q)^{k-m} q & \text{si } k < \eta \\ \binom{k-1}{m-1} q^{m-1} (1-q)^{k-m} q + \sum_{l=0}^{m-1} \binom{\eta}{l} q^l (1-q)^{\eta-l} & \text{si } k = \eta \end{cases} \quad (4.6)$$

són similars, i corresponen a la probabilitat que es necessiten calcular k distàncies (o siga, k verificacions de proximitat/llunyania) per a declarar el central *corrupte* o *no corrupte*, suposant (la qual cosa es desconeix) que el píxel central pugui no ser sorollós o sorollós, respectivament. Centrem-nos ara en 4.5, on se suposa el central no sorollós.

La probabilitat que en el píxel k es declare el central *no corrupte* (amb $4 \leq k \leq 24$, i.e. $m \leq k \leq \eta$) ve donada pel fet que en el píxel $k - 1$ han d'haver-se trobat 3 píxels propers al central i el píxel k ha de ser també proper al central. Aquesta probabilitat és

$$\binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} r$$

D'aquesta manera, en [105], per a declarar el central com *corrupte* s'ha d'arribar a estudiar l'últim píxel 24, doncs per al cas $k = \eta = 24$ cal afegir tots els casos en que hi haja menys de 4 veïns, que són:

$$\binom{24}{0} r^0 (1-r)^{24} + \binom{24}{1} r^1 (1-r)^{23} + \dots + \binom{24}{3} r^3 (1-r)^{21}$$

Així doncs, en general es te

$$f_1(k) = \begin{cases} \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} r, & \text{si } m \leq k < \eta \\ \binom{\eta-1}{m-1} r^{m-1} (1-r)^{\eta-m} r + \sum_{l=0}^{m-1} \binom{\eta}{l} r^l (1-r)^{\eta-l}, & \text{si } k = \eta \end{cases} \quad (4.7)$$

Atenent a l'esquema anterior, altra manera d'obtenir $f_1(24)$ (és a dir $f_1(\eta)$) és la següent.

Continuant en l'exemple en concret, l'algorisme arriba a estudiar el píxel 24 si en el píxel 23 ha detectat solament 0,1,2 o 3 propers al central, la probabilitat de la qual cosa és:

$$\binom{23}{0} r^0 (1-r)^{23} + \binom{23}{1} r^1 (1-r)^{22} + \binom{23}{2} r^2 (1-r)^{21} + \binom{23}{3} r^3 (1-r)^{20}$$

per tant en general l'expressió 4.7 també pot escriure's com

$$f_1(k) = \begin{cases} \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} r, & \text{si } k = m, m+1, \dots, \eta-1 \\ \sum_{l=0}^{m-1} \binom{\eta-1}{l} r^l (1-r)^{\eta-1-l}, & \text{si } k = \eta \end{cases} \quad (4.8)$$

L'inconvenient d'aquesta expressió és que no diferencia el sumatori de probabilitats que declara el píxel central com *corrupte*.

Per a veure que la nostra expressió 4.8 coincideix amb l'expressió 4.5 de Smolka, és suficient veure que coincideixen al terme $k = \eta$. Per això, multiplicarem el terme citat en 4.8 per $r + (1-r)(=1)$ i s'obté:

$$\binom{\eta-1}{0} r^0 (1-r)^{\eta-1} \cdot r + \dots + \binom{\eta-1}{m-2} r^{m-2} (1-r)^{\eta-m+1} \cdot r + \quad (a)$$

$$\binom{\eta-1}{m-1} r^{m-1} (1-r)^{\eta-m} \cdot r + \quad (b)$$

$$\binom{\eta-1}{0} r^0 (1-r)^{\eta-1} \cdot (1-r) + \quad (c)$$

$$\binom{\eta-1}{1} r^1 (1-r)^{\eta-2} \cdot (1-r) + \dots + \binom{\eta-1}{m-1} r^{m-1} (1-r)^{\eta-m} \cdot r \quad (d)$$

(4.9)

Els termes estan escrits en ordre, però situats i etiquetats de manera estratègica per a sumar els termes i obtenir com a expressió (b)+(c)+((a)+(d)). Així obtenim:

$$\binom{\eta-1}{m-1} r^m (1-r)^{\eta-m} + \quad (b)$$

$$\binom{\eta-1}{0} r^0 (1-r)^\eta + \quad (c)$$

$$\binom{\eta}{1} r (1-r)^{\eta-1} + \binom{\eta}{2} r^2 (1-r)^{\eta-2} + \dots + \quad (4.10)$$

$$\binom{\eta}{m-1} r^{m-1} (1-r)^{\eta-m+1} \quad (a) + (d)$$

i tenint en compte que $\binom{\eta-1}{0} = \binom{\eta}{0}$, l'anterior expressió pot escriure's:

$$\binom{\eta-1}{m-1} r^{m-1} (1-r)^{\eta-m} \cdot r + \sum_{l=0}^{m-1} \binom{\eta}{l} r^l (1-r)^{\eta-l} \quad (4.11)$$

que coincideix amb el terme η -èssim de l'expressió de Smolka 4.5.

El nostre algorisme, descrit a la Secció 4.3 està dissenyat de manera que ature el procés de recerca quan trobe els m membres del *peer group* o quan no siga possible arribar als m membres requerits, a diferència del de Smolka. En el cas d'estudi que venim desenvolupant en finestres 5×5 , això serà en $k = 21$ ($\eta - m + 1$) si es tenen zero veïns, en $k = 22$ ($\eta - m + 2$) amb 1 veí, en $k = 23$ ($\eta - m + 3$) amb 2 veïns. Per últim s'aturarà a l'últim píxel ($k = 24$) tant si s'ha arribat al 23 amb 3 veïns, sent indiferent el que passe al píxel 24 (en efecte si és proper s'atura i diagnostiquem el central com *no corrupte* i si no és proper el diagnostiquem com *corrupte*).

Les probabilitats que l'algorisme declare el píxel central *corrupte* o *no corrupte*, en aquests casos $21 \leq k \leq 24$ ($\eta - m + 1 \leq k \leq \eta$) són:

$$f_1(21) = \binom{20}{3} r^3 (1-r)^{17} \cdot r + \binom{21}{0} r^0 (1-r)^{21} \quad (\text{si } k = 21, k = \eta - m + 1)$$

$$f_1(22) = \binom{21}{3} r^3 (1-r)^{18} \cdot r + \binom{22}{1} r^1 (1-r)^{21} \quad (\text{si } k = 22, k = \eta - m + 2)$$

$$f_1(23) = \binom{22}{3} r^3 (1-r)^{19} \cdot r + \binom{23}{2} r^2 (1-r)^{21} \quad (\text{si } k = 23 = \eta - 1)$$

$$f_1(24) = \binom{23}{3} r^3 (1-r)^{20} \cdot r \quad (\text{si } k = 24 = \eta)$$

(4.12)

de manera general es té:

$$f_1(k) = \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} \cdot r$$

(si $k = m, m+1, \dots, \eta - m$)

$$f_1(k) = \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} \cdot r + \binom{\eta - m + 1}{0} r^0 (1-r)^{\eta - m + 1}$$

(si $k = \eta - m + 1$)

$$f_1(k) = \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} \cdot r + \binom{\eta - m + 2}{1} r^1 (1-r)^{\eta - m + 1}$$

(si $k = \eta - m + 2$)

⋮

$$f_1(k) = \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} \cdot r + \binom{\eta - 1}{m-2} r^{m-2} (1-r)^{\eta - m + 1}$$

(si $k = \eta - 1$)

$$f_1(\eta) = \binom{\eta - 1}{m-1} r^{m-1} (1-r)^{\eta - m} \cdot r + \binom{\eta - 1}{m-1} r^{m-1} (1-r)^{\eta - m + 1}$$

(si $k = \eta$)

(4.13)

L'expressió de $f_1(k)$ per als casos en que $k = \eta - m + 1, \eta - m + 2, \dots, \eta - 1$ pot escriure's de la següent forma:

$$f_1(k) = \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} \cdot r + \binom{k}{i-1} r^{i-1} (1-r)^{\eta-m+i} \quad (4.14)$$

denotant $k = \eta - m + i$, on $i = 0, 1, \dots, m - 1$.

No obstant, això és preferible que l'expressió $f_1(\eta)$ s'escriu en forma de dos sumands, a l'igual que en [105], de manera que el primer siga la probabilitat de declarar el píxel central com *no corrupte*, i el segon la probabilitat de declarar-lo com *corrupte*. Aquesta expressió és:

$$\begin{aligned} f_1(\eta) &= \binom{\eta-1}{m-1} r^{m-1} (1-r)^{\eta-m} \cdot r + \binom{\eta-1}{m-1} r^{m-1} (1-r)^{\eta-m} \cdot (1-r) = \\ &= \binom{\eta-1}{m-1} r^m (1-r)^{\eta-m} + \binom{\eta-1}{m-1} r^{m-1} (1-r)^{\eta-m+1} \end{aligned} \quad (4.15)$$

D'aquest mode, els sumands de la dreta de totes les expressions de $f_1(k)$ per a $k \geq \eta - m + 1$, es corresponen amb la probabilitat de declarar el píxel central com *corrupte* per a cada $k = \eta - m + 1, \dots, \eta$, sense necessitat d'arribar sempre a l'últim píxel η .

Llavors, si suposem que el píxel central és no corrupte, la probabilitat que siguin necessàries k distàncies ($k = m, \dots, \eta$) per a declarar el píxel central com a corrupte o no corrupte (diagnosticar el píxel central) és

$$f_1(k) = \begin{cases} \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} r, & \text{si } k = m, m+1, \dots, \eta - m \\ \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} r + \binom{k-1}{i-1} r^{i-1} (1-r)^{\eta-m} (1-r), & \text{si } k = \eta - m + i, i = 1, 2, \dots, m \end{cases} \quad (4.16)$$

La mateixa probabilitat $f_2(k)$, sota la condició que el píxel central és corrupte s'obté reemplaçant r per q en la expressió (4.8).

La segona part de la suma de $f_1(k)$ en (4.16)

$$\binom{k-1}{i-1} r^{i-1} (1-r)^{\eta-m} (1-r), \quad k = \eta - m + i, \quad i = 1, \dots, m$$

correspon a la probabilitat que el píxel central es diagnostique com (provisionalment) corrupte després del càlcul de k distàncies, sota la condició que no

és corrupte. La mateixa interpretació és certa per als termes corresponents de $f_2(k)$.

A la primera fase del primer pas de l'algorisme, el nombre mitjà de distàncies necessàries per a diagnosticar el píxel central en la finestra W és

$$s'(p) = (1-p) \cdot \sum_{k=m}^{\eta} f_1(k) \cdot k + p \cdot \sum_{k=m}^{\eta} f_2(k) \cdot k$$

i tenint en compte que si diagnosticuem el píxel central com no corrupte, altres m píxels es declaren com no corruptes també, el nombre mitjà de distàncies per a diagnosticar un píxel en una finestra és

$$s_1^*(p) = \frac{s'(p)}{(1-p) \cdot (m+1) + p}$$

A la segona fase del primer pas, com es raona a la Secció 4.3, l'algorisme és capaç de classificar el píxel central i “una mitjana” de i ($i \in [0, m]$) píxels nous quan el píxel central es diagnostica com no corrupte en una finestra, llavors el nombre mitjà de distàncies necessàries per a diagnosticar un píxel a la fase 2 de l'anàlisi és

$$s_2^*(p) = \frac{s'(p)}{(1-p) \cdot (i+1) + p}$$

A la primera fase del procés de detecció, l'algorisme analitza $\frac{N_1 \cdot N_2}{\eta + 1}$ finestres, i diagnosticarà 1 o $m+1$ píxels en cada finestra quan el píxel central es declare provisionalment corrupte o no corrupte, respectivament. Així, el nombre mitjà de distàncies necessàries per a diagnosticar un píxel a l'anàlisi complet és

$$s^*(p) = s_1^*(p) \cdot \left(\frac{(1-p) \cdot (m+1) + p}{\eta + 1} \right) + s_2^*(p) \cdot \left(1 - \frac{(1-p) \cdot (m+1) + p}{\eta + 1} \right)$$

Per exemple, per a valors de $p = 10\%$, $\eta = 8$, $r = 0.8$, $q = 0.01$, obtenim:

	$s^*(p)$ ($m = 2$)	$s^*(p)$ ($m = 3$)
$i = 0$	2.365	2.785
$i = 1$	1.400	1.675
$i = 2$	1.056	1.279
$i = 3$		1.076

Per altra part, la probabilitat que un píxel siga detectat com a corrupte és

$$t(p) = (1-p) \sum_{j=0}^{m-1} \binom{\eta - m + j}{j} r^j (1-r)^{\eta - m + 1} +$$

$$p \sum_{j=0}^{m-1} \binom{\eta - m + j}{j} q^j (1-q)^{\eta - m + 1}$$

Per als valors de $p = 10\%$, $\eta = 8$, $m = 3$, $r = 0.8$, $q = 0.01$, tenim que $t(p) = 0.1011$ i, en les mateixes condicions, per a $m = 2$ tenim que $t(p) = 0.0998$, la qual cosa confirma que aquest estimador és molt proper en ambdós casos a la intensitat de soroll p .

Tenint en compte que el filtre VMF necessita calcular $\binom{\eta+1}{2}$ distàncies per a filtrar un píxel, si incloem el procés de filtratge, com es comenta al final de la Secció 4.3, llavors tenim que el número mitjà de distàncies necessàries per a filtrar un píxel és

$$c_d^*(p) = (1 - t(p)) \cdot s^*(p) + t(p) \cdot \left(s^*(p) + \binom{\eta}{2} \right) = s^*(p) + t(p) \cdot \binom{\eta}{2}$$

ja que en el procés de filtratge, el píxel central es considera corrupte, i s'exclou del càlcul.

Ara, seguint les nostres experiències, podem prendre com una aproximació sòlida en l'expressió $c_d^*(p)$, $i = 0.5$ quan $m = 2$ i $i = 1$ quan $m = 3$.

Al procés de detecció descrit en [105], el número mitjà de distàncies per filtrar un píxel es representa per $c_d(p)$ i el seu valor per a $m = 3$, és $c_d(p) = 7.004$, i per a $m = 2$ és $c_d(p) = 5.844$.

La taula següent mostra els valors comparatius.

m	$c_d(p)$	$c_d^*(p)$
2	5.844	4.528
3	7.004	4.506

Per al cas del filtre VMF, el nombre de distàncies necessàries $c_f(p)$ per al procés de filtratge és constant i pren el valor de $c_f(p) = \binom{\eta+1}{2}$.

Com s'observa a la taula anterior i deduït dels nostres estudis experimentals, el guany en termes de la reducció del càlcul de distàncies és d'un 22.5%, dada que es correspon amb l'obtinguda de les nostres experimentacions, on s'obté una reducció del temps computacional al voltant del 20 al 30%, depenent de la intensitat de soroll.

Finalment, introduïm les funcions

$$\gamma_1^m(p) = \frac{c_d(p)}{c_d^*(p)}$$

i

$$\gamma_2^m(p) = \frac{c_f(p)}{c_d^*(p)}$$

per a quantificar el guany computacional de velocitat per al filtre FMFGF comparant-lo amb els filtres PGF i VMF respectivament. Les Figures 4.17 i 4.18 mostren els valors comparatius en termes de la intensitat de soroll p .

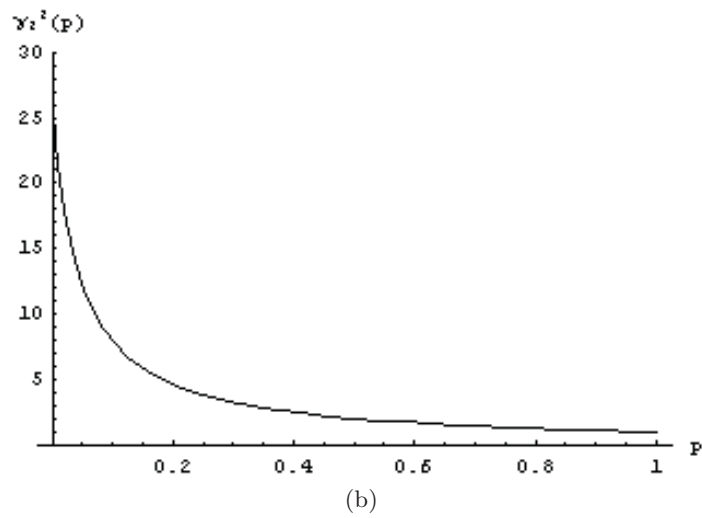
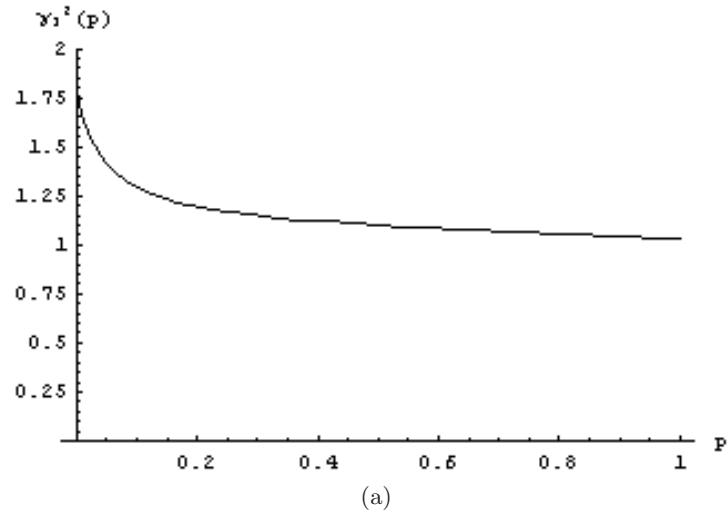


Figura 4.17. Comparativa de velocitat computacional per a $m = 2$: (a) FMPGF versus PGF, (b) FMPGF versus VMF.

4.6 Conclusions

En aquest Capítol, s'introdueix un nou filtre per a eliminació de soroll impulsu basat en una redefinició del concepte de *peer group* fent servir una mètrica fuzzy. El disseny presentat millora significativament els filtres vectorials clàssics i presenta un rendiment similar als filtres vectorials recentment publicats a la literatura, obtenint una relació adequada entre reducció de soroll, preservació dels detalls de la imatge i eficiència computacional. La

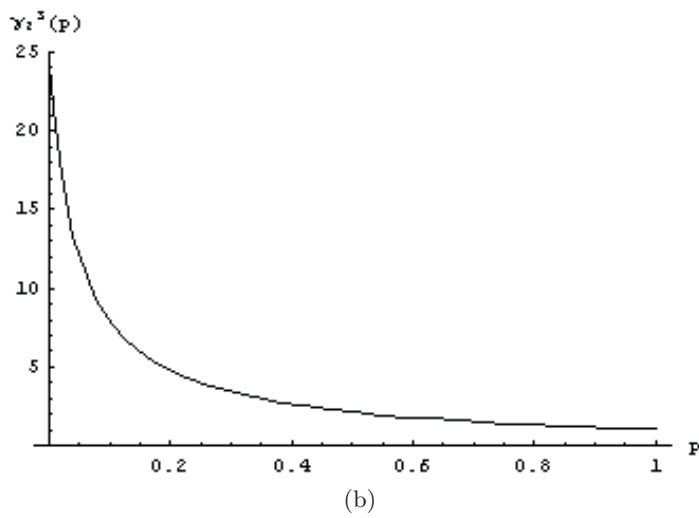
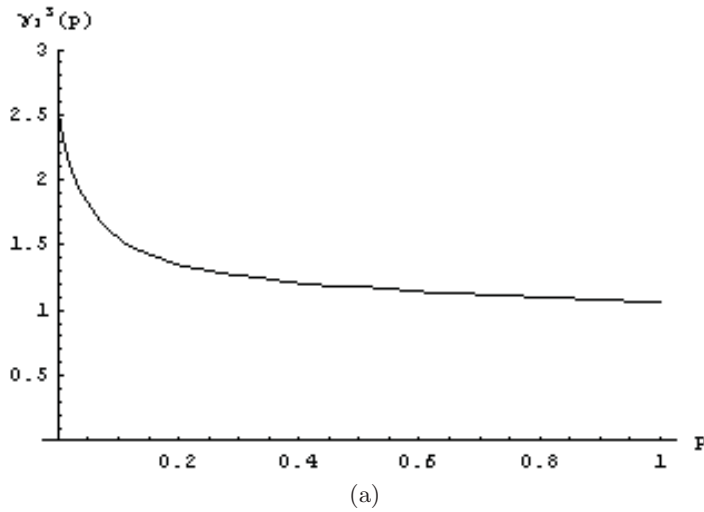


Figura 4.18. Comparativa de velocitat computacional per a $m = 3$: (a) FMPGF versus PGF, (b) FMPGF versus VMF.

complexitat computacional del filtre proposat s'ha demostrat que és menor que les presentades pels filtres vectorials clàssics i els filtres vectorials similars basats en el concepte de *peer group*.

5 Nou Mètode per a la Detecció i Eliminació Ràpida de Soroll Impulsiu amb Mètriques Fuzzy

Joan-Gerard Camarena, Valentín Gregori, Samuel Morillas and Guillermo Peris-Fajarnés “New Method for Fast Detection and Removal of Impulsive Noise Using Fuzzy Metrics”, *Lecture Notes in Computer Science* vol. 1, pp. 359-369 (2006). Proceedings of the *International Conference on Image Analysis and Recognition ICIAR 2006*

Resum

En aquest capítol es presenta un nou mètode per la detecció ràpida de soroll impulsiu en imatges en color. S’analitza el veïnat d’un píxel central amb una mètrica fuzzy per a la detecció ràpida de píxels sorollosos fent servir el concepte de *peer group*. Després, es proposa un filtre basat en un esquema de substitució entre el Filtre de Mitjana Aritmètica (AMF) i l’operació identitat. El filtre proposat aconsegueix un bon balanç entre l’eliminació de soroll i la conservació dels detalls de la imatge, millorant significativament els filtres clàssics vectorials. La present proposta és més ràpida que les tècniques de filtratge de substitució recents basades en conceptes similars, presentant tot i això un rendiment competitiu en quant a termes de qualitat.

5.1 Introducció

Fins l’actualitat s’han proposat certs filtres vectorials per a imatges en color que prenen avantatge entre la correlació entre els canals de la imatge en color i la teoria estadística robusta. Filtres ben coneguts d’aquesta família són el *filtre de mediana vectorial* (VMF) [4], el *filtre direccional bàsic vectorial* (BVDF) [114], o el *filtre de distància direccional* (DDF) [47]¹.

Els filtres vectorials mencionats anteriorment presenten un rendiment robust però tendeixen a difuminar vèrtexs i detalls de la imatge. En tal de superar aquest inconvenient els esquemes de substitució s’han fet servir pel seu adequat rendiment i simplicitat computacional. El esquema de substitució milloren seleccionant un conjunt de píxels de la imatge per a filtrar-los

¹ Pot trobar-se informació més extensa d’aquests filtres i les seues propietats en la visió general feta en [63]

mentre que deixen la resta de píxels invariants. S'han proposat fins la data una sèrie de mètodes per a seleccionar els píxels sorollosos basats en les estadístiques locals de veïnats [1, 11, 62, 67, 68, 69, 73, 88, 102, 104, 105, 106].

El concepte de *peer group* amb mètriques clàssiques es fa servir a [105, 106] per a detectar píxels sorollosos. En aquest capítol, prendrem l'anomenat concepte de *peer group* però fent servir una mètrica fuzzy ja que s'ha provat que dóna millors resultats que les mètriques clàssiques per a filtratge de soroll impulsiu [87, 88]. El concepte de *peer group fuzzy* proposat es fa servir per a definir un filtratge per substitució amb les bases dels treballs mencionats anteriorment.

L'objectiu del present capítol és buscar un mètode ràpid i competitiu, basant-se en el mètode del capítol anterior. En aquest mètode la detecció de soroll, a diferència de tècniques anteriors en les que hi havia dues fases, s'efectua en una sola fase, aconseguint amb això, conjuntament amb una adequada tècnica ràpida de detecció basada en el concepte de *peer group fuzzy*, una gran millora computacional tenint en compte que la qualitat del filtratge es competitiva.

Aquest filtre és més ràpid que els presentats en [105, 106], mostrant un rendiment competitiu respecte dels filtres de substitució introduïts recentment i millorant-los en certs casos. Es mostren els resultats experimentals per a la comparació de rendiments per veure que l'aproximació presentada millora significament els filtres vectorials clàssics.

El Capítol s'organitza com segueix. La Secció 5.2 mostra detalladament l'algorisme de filtratge proposat. L'estudi experimental i diverses comparatives davant de filtres vectorials, tant clàssics com recents, es presenten a la Secció 5.3. Per acabar, la Secció 5.5 presenta les conclusions.

5.2 Detecció i Eliminació de Píxels Corruptes

Una imatge a l'espai de color RGB es representa normalment com un *array* multidimensional $N_1 \times N_2 \times 3$, on cada píxel \mathbf{x}_i ($i \in N_1 \times N_2$) és un vector de tres components de valors enters dins l'interval $[0,255]$. Ara, atenent al concepte de proximitat que ens facilita la mètrica fuzzy M vista en el capítol anterior donada per

$$M(\mathbf{x}_i, \mathbf{x}_j) = \prod_{l=1}^3 \frac{\min\{x_i(l), x_j(l)\} + K}{\max\{x_i(l), x_j(l)\} + K} \quad (5.1)$$

per a un píxel central \mathbf{x}_i en una finestra W de grandària $n \times n$, $n = 2w + 1$, $w = 1, 2, \dots$ i fixat $d \in]0, 1]$, representem per $\mathcal{P}(\mathbf{x}_i, d)$ el veïnat de \mathbf{x}_i contingut en W

$$\{\mathbf{x}_j \in W : M(\mathbf{x}_i, \mathbf{x}_j) \geq d\}$$

Taula 5.1. Filtres estudiats per a la comparativa de rendiment junt a la notació emprada.

Notació	Filtre
VMF	Filtre de Mitjana Vectorial[4]
DDF	Filtre de Distància Direccional [47]
BVDF	Filtre Basic Vectorial Direccional[114]
FIVF	Filtre Impulsiu Vectorial Ràpid[88]
PGF	Filtre de Peer Group [105, 106]
FMNF	Filtre del Veïnat amb Mètrica Fuzzy

això és, $\mathcal{P}(\mathbf{x}_i, d)$ és el conjunt de píxels de la finestra de filtratge W la distància fuzzy dels quals a \mathbf{x}_i no és menor a d . És obvi que $\mathcal{P}(\mathbf{x}_i, d)$ és un conjunt no buit per a cada \mathbf{x}_i , ja que $\mathbf{x}_i \in \mathcal{P}(\mathbf{x}_i, d)$.

Ara, representarem per $\mathcal{P}(\mathbf{x}_i, m, d)$ un subconjunt de $\mathcal{P}(\mathbf{x}_i, d)$ amb $m + 1$ elements, i seguint els criteris de [105] és un *peer group* de \mathbf{x}_i en W , per a la mètrica fuzzy M i per un $d \in]0, 1[$ fix. En particular, si $\mathcal{P}(\mathbf{x}_i, d)$ té $c + 1$ elements, llavors pot denotar-se per $\mathcal{P}(\mathbf{x}_i, c, d)$.

L'algorisme proposat de detecció i eliminació de soroll impulsiu, anomenat a partir d'ara *Filtre del Veïnat amb Mètrica Fuzzy* (FMNF), és una modificació de l'explicat a [105], i està basat en el concepte de *peer group* $\mathcal{P}(\mathbf{x}_i, m, d)$ definit anteriorment, involucrant-hi la mètrica fuzzy amb la que treballem al llarg de la Tesi, i funciona en dos passos com es detalla a continuació.

Al primer pas els píxels de la imatge són *diagnosticats*. L'algorisme pren finestres de grandària $n \times n$, $n = 2w + 1$, $w = 1, 2 \dots$ per a un píxel central \mathbf{x}_i i donat $d \in]0, 1[$ es calcula el conjunt $\mathcal{P}(\mathbf{x}_i, d)$. A continuació, per a un paràmetre $m \in \{0, \dots, n^2 - 1\}$, si podem trobar un *peer group* $\mathcal{P}(\mathbf{x}_i, m, d)$, llavors tots els píxels de $\mathcal{P}(\mathbf{x}_i, d)$ es diagnostiquen com píxels *no corruptes*.

Vegeu que, en aquestes condicions, fent servir l'axioma FM4 de la mètrica fuzzy, podem garantir que la distancia fuzzy entre dos punts qualssevol del conjunt $\mathcal{P}(\mathbf{x}_i, d)$ serà major o igual a d^2 , i per tant, amb una elecció encertada del valor de d , aquells píxels en el conjunt mencionat poden declarar-se com *no corruptes* amb bastant seguretat. Aquest fet redueix dràsticament el nombre de càlculs a efectuar sense disminuir el rendiment del procés de detecció com podrà veure's a la Secció 5.3. La resta de píxels de la finestra que no pertanyen al conjunt $\mathcal{P}(\mathbf{x}_i, d)$ es declaren com *no diagnosticats*. Si en el conjunt $\mathcal{P}(\mathbf{x}_i, d)$ per al píxel central \mathbf{x}_i conté menys de $m + 1$ elements, llavors \mathbf{x}_i es declara *provisionalment corrupte* i cada píxel de la finestra es declara com *no diagnosticat*.

Aquest procés es repeteix per a tots els píxels en la imatge, evitant aquells que ja s'hagen declarat com a *no corruptes*. Adoneu-se que certs píxels *provisionalment corruptes* poden diagnosticar-se com a *no corruptes* posteriorment si aquests formen part d'una finestra en la qual s'ha pogut fer el conjunt

$\mathcal{P}(\mathbf{x}_i, d)$ amb almenys $m + 1$ elements. Finalment, la resta de píxels que eren *provisionalment corruptes* es declaren com *corruptes*. A l'Algorisme 4 es veu una descripció detallada en format algorísmic.

Al segon pas els píxels *corruptes* són substituïts efectuant l'operació de filtratge. Cada píxel *corrupte* es reemplaça per la sortida del filtre de mitjana vectorial (AMF) sobre (imitant a [105]) els píxels veïns *no corruptes* del píxel a substituir dins d'una finestra $n \times n$. Adonar-se que un píxel *corrupte* pot no tindre cap veí *no corrupte*, amb la qual cosa la finestra ha d'augmentar-se de grandària, fins que almenys es dispose d'un píxel *no corrupte* dins de la finestra. Aquest procés es veu a l'Algorisme 5. La tècnica del AMF és la que s'ha decidit aplicar per a la seua simplicitat computacional, encara que podrien haver-se aplicat de la mateixa forma altres tècniques similars de filtratge adequades com el VMF, en les condicions anteriors.

Algorisme 4: FMNF - Pas 1 Detecció de Píxels Sorollosos

Entrades: $m, d, n, \text{Imatge_Original}$

Sortides : Matriu_Soroll

```

1 Obtenir  $N_1, N_2$ , dimensions de  $\text{Imatge\_Original}$ ;
2 Per A  $i = 1$  fins  $N_1$  Fer
3   Per A  $j = 1$  fins  $N_2$  Fer
4     Si  $\text{Matriu\_Soroll}_{i,j} == 0$  Llavors
5       | Passar al píxel següent;
6     Si no
7       | Construir finestra  $W$  amb  $\mathbf{x}_{i,j}$  com a píxel central;
8       | Calcular  $\mathcal{P}(\mathbf{x}_{i,j}, d)$ ;
9       | Si  $\#\mathcal{P}(\mathbf{x}_{i,j}, d) \geq m + 1$  Llavors
10        |  $\forall \mathbf{x}_j \in \mathcal{P}(\mathbf{x}_{i,j}, d)$ ,  $\mathbf{x}_j$  es declara com no corrupte ;
11        | Si no
12          |  $\mathbf{x}_{i,j}$  es declara com corrupte provisionalment ;
13          |  $\forall \mathbf{x}_k \in W - \mathbf{x}_{i,j}$ ,  $\mathbf{x}_k$  es declara com no diagnosticat ;
14        | FiSi
15        | /* corrupte=1,no_diagnosticat=1,no corrupte=0 */
16        | Transcriure  $W$  a  $\text{Matriu\_Soroll}$ ;
17      | FiSi
18    FiPer
19  FiPer

```

Algorisme 5: FMNF - Pas 2 Aplicació de la Tècnica de Filtratge per Substitució

Entrades: n , *Imatge_Original*, *Matriu_Soroll*

Sortides : *Imatge_Filtrada*

```
1 Obtenir  $N_1, N_2$ , dimensions de Imatge_Original;  
2 Per A  $i = 1$  fins  $N_1$  Fer  
3   Per A  $j = 1$  fins  $N_2$  Fer  
4     Si (Matriu_Soroll $_{(i,j)} == 1$ ) Llavors  
5        $W =$  Finestra amb centre  $\mathbf{x}_{(i,j)}$  de talla  $n \times n$ ;  
6       Mentre ( $\#(\text{píxels no corruptes de } W) = 0$ ) Fer  
7          $n = n + 1$ ;  
8          $W =$  Finestra amb centre  $\mathbf{x}_{(i,j)}$  de talla  $n \times n$ ;  
9       FiMentre  
10      Imatge_Filtrada $_{(i,j)} = AMF_{out}$  sobre els píxels no corruptes  
11      de  $W$ ;  
12      Si no  
13      Imatge_Filtrada $_{(i,j)} =$  Imatge_Original $_{(i,j)}$ ;  
14    FiSi  
15  FiPer  
16 FiPer
```

5.3 Estudi Experimental i Comparativa de Rendiment

En aquesta secció, s'han pres en consideració diverses imatges per avaluar el rendiment del filtre FMNF proposat i explicat a la Secció 5.2. El model del soroll impulsiu per al soroll de transmissió definit en [84] s'ha fet servir per afegir soroll a les imatges de Aerial1, Baboon and Lenna (veure la Figura. 5.1). El procés de corrupció del soroll impulsiu o model d'impulsos està descrit a l'Equació 2.2 al Capítol 2.

Com a mesures objectives per avaluar la qualitat del rendiment del filtratge, s'han pres l'*Error Mitjà Absolut* (MAE), el *Ratio de Soroll en Pics de Senyal* (PSNR) i la *Diferència de Color Normalitzat* (NCD) definides al Capítol 2.

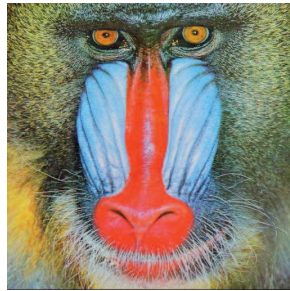
El filtre proposat es confronta davant tant d'alguns filtres clàssics com de filtres recents amb bona habilitat de conservació de detalls, i estan enumerats a la Taula 5.1.

Els resultats experimentals² es veuen a les Taules 5.2-5.4. D'ara endavant, denotarem per p la intensitat de soroll de la imatge, que pot considerar-se com la probabilitat d'aparició del soroll. Nosaltres hem pres per a les nostres experimentacions valors de $p \in \{0.05, 0.1, 0.15, 0.2\}$.

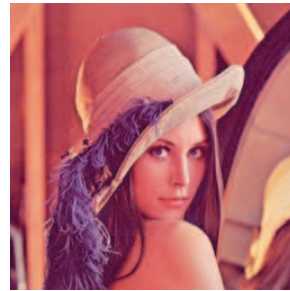
² Obtinguts en un Pentium IV, 3.4 GHz, 512 MB RAM.



(a)



(b)



(c)

Figura 5.1. Imatge de l'Experimentació: (a) Aerial1, (b) Baboon, (c) Lenna.

Podem concloure que els millors resultats del nostre filtre en finestres 3×3 els hem obtingut per a $m = 3, 4$ i $d \in [0.88, 0.94]$ i particularment el valor de $d = 0.92$ pot prendre's com un valor robust, en general. Pot veure's que el filtre proposat millora significativament els filtres vectorial clàssics i el seu rendiment és competitiu respecte als altres esquemes de substitució presos en consideració. Tot i això, com es veu a les Taules 5.2-5.4, el filtre proposat dóna millors resultats en imatges amb molts detalls fins i textures, com les imatges Aerial1 o Baboon front els altres tipus d'imatges. Les Figures 5.2 a 5.5 mostren el rendiment del filtre FMNF en funció del paràmetre d front els filtres VMF i PGF (fent servir el valor de $d = 45$ proposat a [105]) que ara passem a comentar.

A les figures 5.2 i 5.3 es veu l'estudi comparatiu de rendiment objectiu (mesurat amb el PSNR i NCD) i de cost computacional del FMNF davant el PGF i el VMF en la imatge Aerial contaminada amb 10% de soroll impulsiu. Els rendiments tant del VMF com del PGF estan expressats en el millors

Taula 5.2. Comparativa de Rendiment per a la imatge *Aerial1* 150×200 .

p		MAE	PSNR	NCD (10^{-2})	Temps Comp. (T) (seg)	ΔT FMNF vs. millor PGF
0.05	VMF	17.271	19.530	9.204	70.97	40.4%
	DDF	17.059	19.303	8.644	134.16	
	BVDF	21.842	16.884	10.401	78.03	
	FIVF	3.033	23.698	2.565	37.41	
	PGF $_{m=2}$	7.507	21.237	3.334	4.75	
	PGF $_{m=3}$	15.404	17.696	6.840	5.99	
	FMNF	3.343	23.852	2.475	2.83	
0.10	VMF	17.508	19.422	9.357	70.38	37.2%
	DDF	17.441	19.209	8.845	134.09	
	BVDF	22.739	16.612	10.770	77.53	
	FIVF	5.243	21.898	3.936	37.20	
	PGF $_{m=2}$	9.198	20.392	4.249	4.95	
	PGF $_{m=3}$	16.802	17.253	7.761	6.17	
	FMNF	5.210	22.260	3.841	3.11	
0.15	VMF	17.719	19.298	9.514	70.41	33.9%
	DDF	17.892	19.040	9.052	133.56	
	BVDF	23.324	16.400	11.010	77.61	
	FIVF	7.605	20.597	5.373	37.61	
	PGF $_{m=2}$	11.019	19.611	5.318	5.33	
	PGF $_{m=3}$	18.321	16.791	8.909	6.33	
	FMNF	7.753	20.713	5.366	3.52	
0.20	VMF	18.031	19.158	9.742	70.39	29.7%
	DDF	18.598	18.827	9.364	133.39	
	BVDF	24.343	16.197	11.435	77.61	
	FIVF	10.371	19.549	6.589	37.25	
	PGF $_{m=2}$	13.087	18.774	6.617	5.42	
	PGF $_{m=3}$	19.608	16.448	10.252	6.61	
	FMNF	10.374	19.393	6.957	3.81	

resultats d'aquests filtres, i és per això que apareixen com una línia horitzontal. El PSNR és un paràmetre que representa la capacitat de l'algorisme de filtratge per a l'eliminació de soroll, sent millor a valors més alts. D'altra banda el NCD és un paràmetre que mesura la qualitat visual a la inspecció humana de la imatge filtrada, sent millor a valors més baixos. Com s'observa en aquestes figures, els valors adequats per al nostre algorisme del paràmetre distància d són fins un valor de 0.91 moment a partir del qual el valor comença a degradar el resultat. El temps computacional s'observa que sempre és menor, amb variacions des de 2 segons menys en el millor dels casos (al voltant del 40% en el millor dels casos i amb un 30% per terme mig).

Taula 5.3. Comparativa de Rendiment per a la imatge *Baboon* 256×256 .

p		MAE	PSNR	NCD (10^{-2})	Temps Comp. (T) (seg)	ΔT FMNF vs. millor PGF
0.05	VMF	10.878	23.116	4.496	156.28	24.9%
	DDF	10.708	23.029	4.201	315.72	
	BVDF	11.891	22.065	4.548	171.38	
	FIVF	1.698	27.055	1.555	81.31	
	PGF _{$m=2$}	2.678	26.931	1.062	7.72	
	PGF _{$m=3$}	4.874	24.483	1.857	10.06	
	FMNF	2.140	27.328	1.180	5.80	
0.10	VMF	10.962	23.073	4.541	158.45	20.0%
	DDF	10.920	22.952	4.311	316.86	
	BVDF	12.147	21.930	4.661	173.02	
	FIVF	2.891	25.519	2.258	81.47	
	PGF _{$m=2$}	3.543	26.007	1.439	8.24	
	PGF _{$m=3$}	5.714	23.962	2.220	10.94	
	FMNF	3.562	25.656	1.763	6.59	
0.15	VMF	11.083	22.999	4.607	155.88	17.3%
	DDF	11.164	22.838	4.442	318.19	
	BVDF	12.585	21.723	4.853	173.92	
	FIVF	4.012	24.440	2.963	81.33	
	PGF _{$m=2$}	4.386	25.275	1.867	8.97	
	PGF _{$m=3$}	6.570	23.486	2.661	11.53	
	FMNF	4.882	24.589	2.331	7.42	
0.20	VMF	11.205	22.919	4.682	155.88	15.5%
	DDF	11.401	22.740	4.569	328.14	
	BVDF	12.903	21.511	5.001	171.23	
	FIVF	5.329	23.977	3.024	81.09	
	PGF _{$m=2$}	5.512	24.239	2.525	9.63	
	PGF _{$m=3$}	7.530	22.952	3.229	12.20	
	FMNF	6.306	23.553	3.032	8.14	

A les figures 5.4 i 5.5 es veu el mateix estudi però amb la imatge Baboon contaminada en un 20% de soroll impulsiu. Els resultats són bastant similars, però ajustant el paràmetre de distància $d = 0.92$ (valor que finalment hem considerat com robust). La variació en temps computacional varia entre 3 i 1 segons, estant al voltant del 20% de millora.

La millora computacional del filtre PGF respecte al VMF va ser mostrada mitjançant l'estudi de complexitat computacional del PGF. Ara, recordem que en [105] el concepte de *peer group* (amb mètriques clàssiques) es fa servir per diagnosticar solament el píxel central de la finestra corredera, i que aquest procés s'efectua en tots els píxels de la imatge. El nostre algorisme ens permet diagnosticar diversos píxels quan el píxel central és *no corrupte* i lla-

Taula 5.4. Comparativa de Rendiment per a la imatge *Lenna* 256×256 .

p		MAE	PSNR	NCD (10^{-2})	Temps Comp. (T) (seg)	ΔT FMNF vs. millor PGF
0.05	VMF	2.755	32.051	1.781	155.97	9.7%
	DDF	2.764	31.808	1.663	317.50	
	BVDF	2.985	31.382	1.720	173.34	
	FIVF	0.414	35.685	0.482	81.38	
	PGF $_{m=2}$	0.4037	37.886	0.297	6.11	
	PGF $_{m=3}$	0.7386	34.384	0.445	8.02	
	FMNF	0.699	34.483	0.471	5.52	
0.10	VMF	2.874	31.767	1.847	155.89	11.2%
	DDF	2.982	31.325	1.788	321.36	
	BVDF	3.248	30.773	1.869	174.73	
	FIVF	0.743	34.158	0.669	81.17	
	PGF $_{m=2}$	0.696	35.257	0.553	6.78	
	PGF $_{m=3}$	1.072	32.857	0.705	8.74	
	FMNF	1.067	32.580	0.786	6.02	
0.15	VMF	2.946	31.611	1.903	157.80	10.5%
	DDF	3.159	31.006	1.915	316.08	
	BVDF	3.513	30.112	2.035	172.22	
	FIVF	1.031	32.972	0.936	81.09	
	PGF $_{m=2}$	1.081	32.397	0.915	7.36	
	PGF $_{m=3}$	1.442	31.639	0.982	9.45	
	FMNF	1.458	31.343	1.084	6.59	
0.20	VMF	3.073	31.265	1.990	156.52	11.4%
	DDF	3.386	30.524	2.057	314.83	
	BVDF	3.788	29.540	2.196	172.67	
	FIVF	1.350	31.936	1.191	81.03	
	PGF $_{m=2}$	1.527	30.515	1.334	8.06	
	PGF $_{m=3}$	1.896	30.188	1.345	10.28	
	FMNF	1.941	29.965	1.456	7.14	

vors hi haurà molts casos on no serà necessari efectuar el procés de la finestra corredera (veure l'Algorisme 4, línia 4). Amb tot això, el principal avantatge de la tècnica de filtratge proposada és la gran velocitat computacional (veure Taules 5.2-5.4 i Figures 5.2 a 5.2). Per últim, la Taula 5.5 mostra una comparativa de la càrrega computacional entre el filtre PGF $_{m=2}$ (que dona els millors resultats) i el filtre FMNF deduït de les nostres experimentacions amb les imatges anteriors. En aquesta taula, els valors de e_p i e_p^* denoten el nombre mitjà de distàncies calculades que són necessàries per diagnosticar un píxel per als casos dels filtres PGF $_{m=2}$ i el nostre FMNF, respectivament.

Així, suposem una imatge de grandària mitjana (per exemple 200×200 píxels), amb un total de 40000 píxels a diagnosticar contaminada amb un 15%

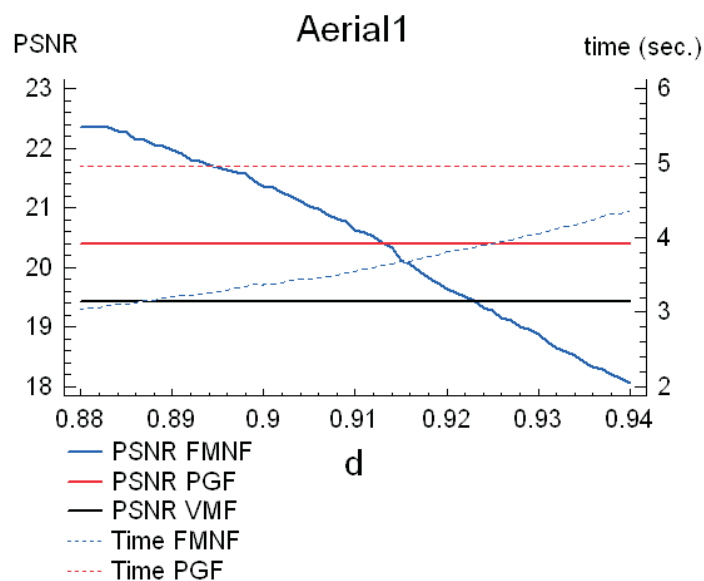


Figura 5.2. Comparativa de PSNR i temps computacional en funció de d a la imatge Aerial1 amb un 10% de soroll impulsiu.

Taula 5.5. Comparativa de la Càrrega Computacional entre els filtres $\text{PGF}_{m=2}$ i FMNF.

p	e_p	e_p^*
0.05	3.52	2.51
0.10	3.82	2.72
0.15	4.15	2.94
0.20	4.46	3.21

de soroll impulsiu. En el filtratge PGF caldran efectuar un total de $4.15 \times 40000 = 166.000$ distàncies mentre que al FMNF seran necessàries $2.94 \times 40000 = 117.600$ distàncies, és a dir al voltant d'un 30% menys distàncies.

A les Figures 5.6-5.8 es veuen les mostres dels detalls de les imatges un cop filtrades per a Aerial amb $p = 0.1$ i Baboon i Lenna amb $p = 0.2$.

5.4 Complexitat Computacional

Seguint el raonament de [105] i del capítol anterior, siga η el nombre de veïns del píxel central en una finestra de filtratge W (en finestres $n \times n$, $\eta = n^2 - 1$).

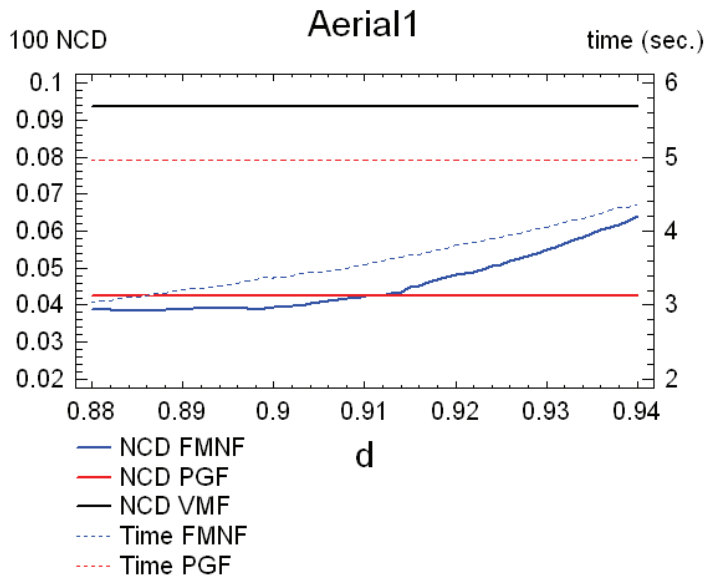


Figura 5.3. Comparativa de NCD i temps computacional en funció de d a la imatge Aerial1 amb un 10% de soroll impulsiu.

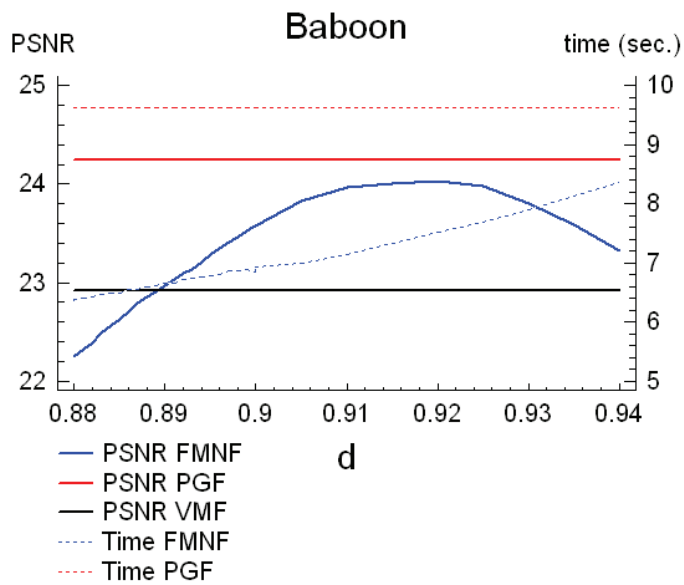


Figura 5.4. Comparativa de PSNR i temps computacional en funció de d a la imatge Baboon amb un 20% de soroll impulsiu.

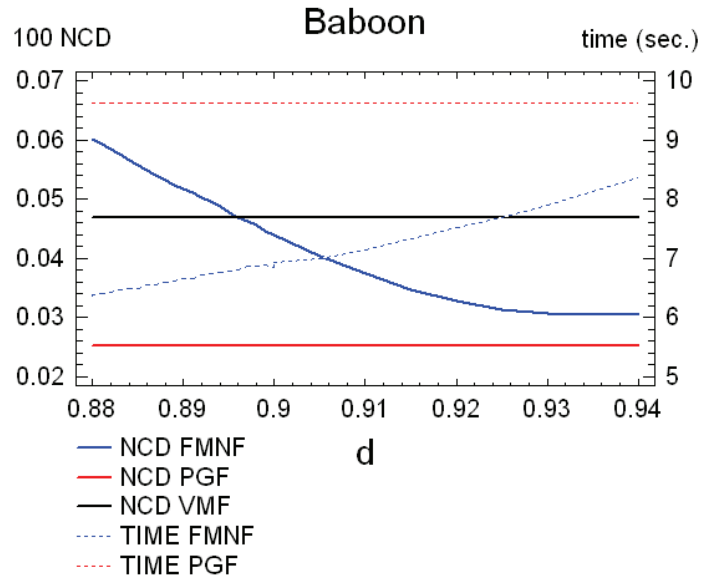


Figura 5.5. Comparativa de NCD i temps computacional en funció de d a la imatge Baboon amb un 20% de soroll impulsiu.



Figura 5.6. Imatges de Sortida: (a) Detall de la imatge Aerial1 original, (b) detall de la imatge Aerial1 amb 10% de soroll impulsiu, (c) sortida del filtre VMF, (d) sortida del filtre DDF, (e) sortida del filtre BVFD, (f) sortida del filtre FIVF, (g) sortida del filtre $PGF_{m=2}$, (h) sortida del filtre $PGF_{m=3}$, (i) sortida del filtre FMNF.

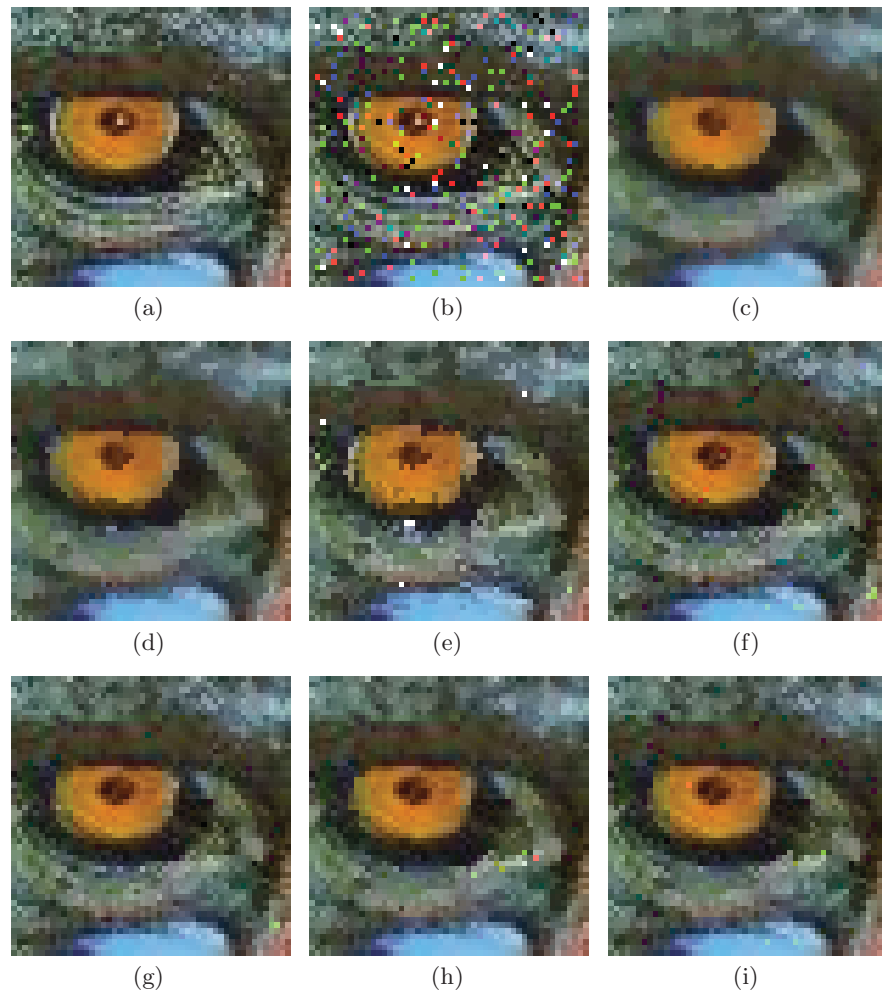


Figura 5.7. Imatges de Sortida: (a) Detall de la imatge Baboon original, (b) detall de la imatge Baboon amb 20% de soroll impulsiu, (c) sortida del filtre VMF, (d) sortida del filtre DDF, (e) sortida del filtre BVFD, (f) sortida del filtre FIVF, (g) sortida del filtre $PGF_{m=2}$, (h) sortida del filtre $PGF_{m=3}$, (i) sortida del filtre FMNF.

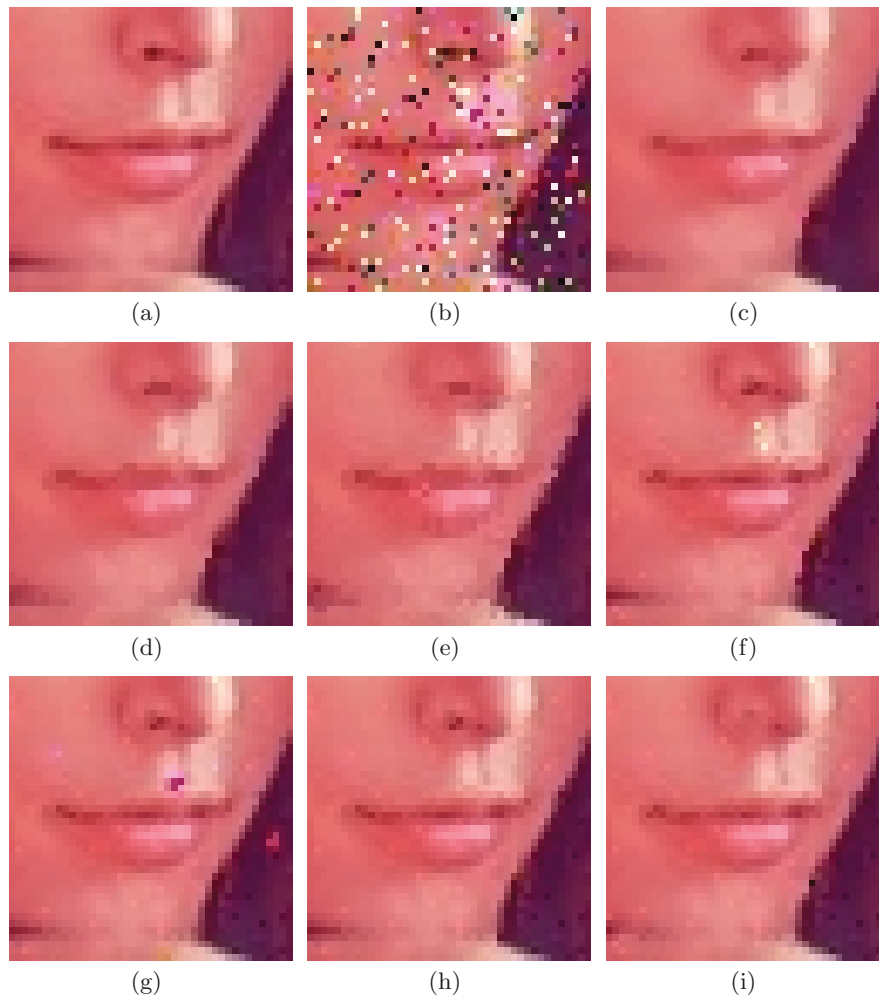


Figura 5.8. Imatges de Sortida: (a) Detall de la imatge original Lenna, (b) Lenna amb 20% de soroll impulsu, (c) sortida del filtre VMF, (d) sortida del filtre DDF, (e) sortida del filtre BVFD, (f) sortida del filtre FIVF, (g) sortida del filtre $PGF_{m=2}$, (h) sortida del filtre $PGF_{m=3}$, (i) sortida del filtre FMNF.

Per trobar quantes distàncies es calculen per terme mitjà en una finestra de diagnòstic W (ja que el càlcul de distàncies és l'operació més demandada), definim les següents variables.

Siguen r i q les probabilitats que un píxel \mathbf{x}_j del veïnat del píxel central \mathbf{x}_i dins de W estiga inclòs o al conjunt de veïns propers (*peer group*) $\mathcal{P}(\mathbf{x}_i, d)$, quan el píxel central és *no corrupte* o *corrupte*, respectivament. Prenent un valor adequat de d , podem suposar que els elements del conjunt $\mathcal{P}(\mathbf{x}_i, d)$ no són corruptes quan \mathbf{x}_i és *no corrupte*, i són corruptes si \mathbf{x}_i és *corrupte*, i per tant, podem prendre $r < 1 - p$ i $q < p$. Les probabilitats r i q depenen de l'estructura de la imatge, l'estadística i intensitat de soroll així com del paràmetre de la distància fuzzy d .

L'algorisme descrit a la Secció 5.2 està dissenyat per aturar el procés de recerca quan no hi ha possibilitat d'arribar al nombre de $m + 1$ membres requerits. Amb tot això, si suposem que el píxel central és *no corrupte*, llavors la probabilitat que siguin necessàries k distàncies ($k = m + 1, \dots, \eta$) per diagnosticar el píxel central com *corrupte* o *no corrupte* és

$$f_1(k) = \begin{cases} \binom{k-1}{i-1} r^{i-1} (1-r)^{\eta-m} (1-r) & \text{si } k = \eta - m + i, \quad i = 1, 2, \dots, m-1 \\ r^m + \binom{k-1}{m-1} r^{m-1} (1-r)^{k-m} (1-r), & \text{si } k = \eta \end{cases} \quad (5.2)$$

La mateixa probabilitat, sota la condició que el píxel central és corrupte, és

$$f_2(k) = \begin{cases} \binom{k-1}{i-1} q^{i-1} (1-q)^{\eta-m} (1-q) & \text{si } k = \eta - m + i, \quad i = 1, 2, \dots, m-1 \\ q^m + \binom{k-1}{m-1} q^{m-1} (1-q)^{k-m} (1-q), & \text{si } k = \eta \end{cases} \quad (5.3)$$

Adoneu-se que la probabilitat que el nombre de distàncies necessàries per a declarar el píxel com *no corrupte* siga igual a k ($k = m, \dots, \eta$), sota la condició que el central és *no corrupte*, és, (veure equació (5.2))

$$\binom{k-1}{m-1} \cdot r^{m-1} \cdot (1-r)^{k-m} \cdot r, \quad \text{per a } k = m, m+1, \dots, \eta$$

D'altra banda, el segon sumand de $f_1(k)$ en (5.2)

$$\binom{k-1}{i-1} r^{i-1} (1-r)^{\eta-m} (1-r), \quad k = \eta - m + i, \quad i = 1, \dots, m$$

representa la probabilitat que el píxel central es diagnostique com *corrupte* (provisionalment) després del càlcul de k distàncies sota la condició que és no corrupte. La mateixa interpretació roman vertadera en termes de $f_2(k)$.

L'algorisme diagnosticarà un o $m+1$ píxels en cada finestra quan el central siga declarat com *provisionalment corrupte* o *no corrupte*, respectivament.

Continuant, el nombre mitjà de distàncies necessari per a diagnosticar el píxel central en una finestra W és

$$s'(p) = (1-p) \cdot \sum_{k=\eta-m+1}^{\eta} f_1(k) \cdot k + p \cdot \sum_{k=\eta-m+1}^{\eta} f_2(k) \cdot k$$

i tenint en compte que en el diagnòstic d'un píxel central *no corrupte*, altres x píxels també es declaren com a *no corruptes*, el nombre mitjà de distàncies necessàries per a diagnosticar un píxel en una finestra és

$$s^*(p) = \frac{s'(p)}{(1-p) \cdot (x+1) + p}$$

Per exemple, donats els valors de $p = 10\%$, $\eta = 8$, $r = 0.8$, $q = 0.01$, $m = 3$, s'obté:

x	0	1	2	3
$s^*(p)$	4.30089	2.26362	1.53603	1.16240
x	4	5	6	7
$s^*(p)$	0.93497	0.78198	0.67201	0.58916

Per altra part, la probabilitat que un píxel siga diagnosticat com a *corrupte* és

$$t(p) = (1-p) \sum_{j=0}^{m-1} \binom{\eta-m+j}{j} r^j (1-r)^{\eta-m+1} + p \sum_{j=0}^{m-1} \binom{\eta-m+j}{j} q^j (1-q)^{\eta-m+1}$$

Per a valors de $p = 10\%$, $\eta = 8$, $m = 3$, $r = 0.8$, $q = 0.01$, tenim que $t(p) = 0.1011$ i, en les mateixes condicions però amb $m = 2$ s'obté $t(p) = 0.0998$, el que confirma que aquest estimador és molt proper a la intensitat de soroll p en ambdós casos.

Prenent que el filtre VMF necessita calcular $\binom{\eta+1}{2}$ distàncies per al filtratge d'un píxel, si l'inclouem en el procés de filtratge, com es comenta a la Secció 5.2, llavors tindrem que el nombre mitjà de distàncies necessàries per al filtratge d'un píxel és

$$c_d^*(p) = (1 - t(p)) \cdot s^*(p) + t(p) \cdot \left(s^*(p) + \binom{\eta}{2} \right) = s^*(p) + t(p) \cdot \binom{\eta}{2}$$

ja que, al procés de filtratge, el píxel central es considera corrupte i per tant s'exclou de la computació.

Per als valors $p = 10\%$, $\eta = 8$, $r = 0.8$, $q = 0.01$, $m = 3$, les següents taules mostren els valors de $c_d^*(p)$ en funció de x .

x	0	1	2	3
$c_d^*(p)$	7.13177	5.0945	4.36691	3.99328
x	4	5	6	7
$c_d^*(p)$	3.76585	3.61286	3.50289	3.42004

Podem prendre el valor $x = 3$ com aproximació als nostres resultats experimentals. Per tant el valor de $c_d^*(p)$ que aproxima les nostres experiències és 3.99328.

Al procés de detecció descrit en [105], el nombre mitjà de distàncies necessàries per al filtratge d'un píxel ve donat per $c_d(p)$ i el seu valor, calculat per a $m = 3$, és $c_d(p) = 7.004$, i per a $m = 2$ és $c_d(p) = 5.844$.

Noteu que, en el cas del VMF, el nombre mitjà de distàncies $c_f(p)$ necessàries per al procés de filtratge és constant i pren el valor

$$c_f(p) = \binom{\eta + 1}{2}.$$

Com s'observa a la taula anterior i deduït dels nostres estudis experimentals, el guany en termes de la reducció del càlcul de distàncies és d'un 32.8%, dada que es correspon amb l'obtinguda de les nostres experimentacions, on s'obté una reducció del temps computacional al voltant del 25 al 35%, dependent de la intensitat de soroll.

Finalment, introduïm les funcions

$$\gamma_1(p) = \frac{c_d(p)}{c_d^*(p)}$$

i

$$\gamma_2(p) = \frac{c_f(p)}{c_d^*(p)}$$

per tal de quantificar el guany de velocitat computacional en el FMNF comparant-se amb els filtres $\text{PGF}_{m=2}$ i VMF, respectivament. La Figura 5.9 mostra la comparativa de resultats en termes de la intensitat de soroll p .

5.5 Conclusions

En aquest Capítol, fem servir el concepte de *peer group* basat en una mètrica fuzzy per a definir un nou filtre de soroll impulsu que és una variació adequada del filtre del *peer group* proposat en [105].

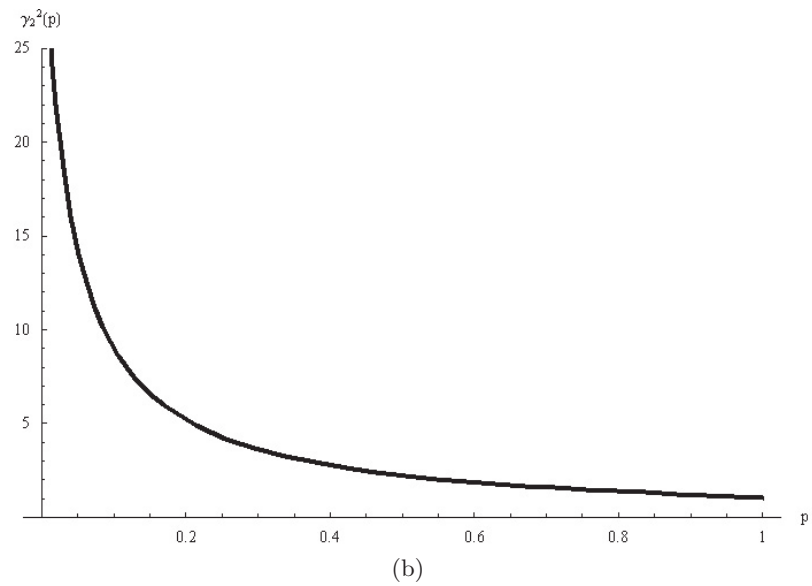
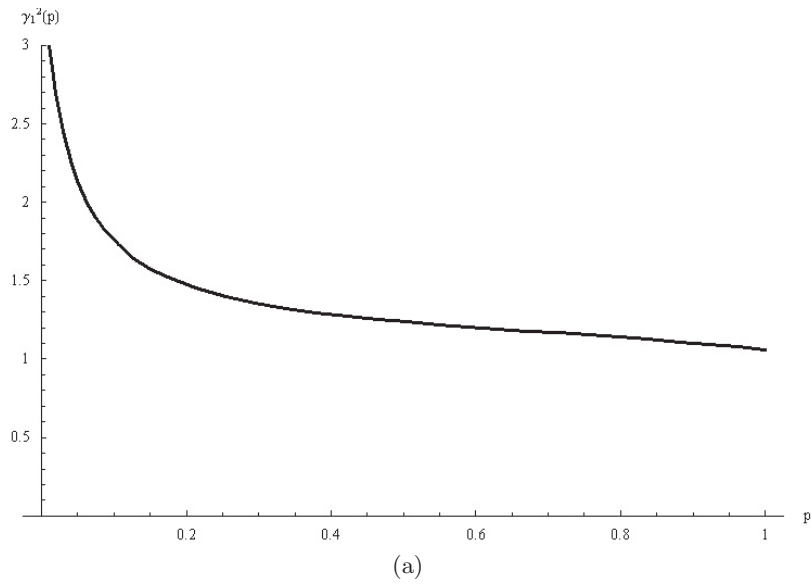


Figura 5.9. Comparativa de velocitat computacional: (a) FMNF versus $\text{PGF}_{m=2}$, (b) FMNF versus VMF

El present disseny millora significativament els filtres vectorials clàssics, presentant un rendiment similar a les tècniques recentment introduïdes vectorials aconseguint un balanç entre la reducció de soroll i la conservació de detalls, particularment en imatges amb molts detalls i textures. La complexitat computacional del filtre proposat és més baixa que el filtre de *peer group* [105], estant basat en un concepte similar.

L'adequació de les mètriques fuzzy per a tasques de processament de imatges es veu confirmat altre cop. A més a més, l'ús de la mètrica fuzzy proposada en substitució de les mètriques clàssiques suposa un guany en l'eficiència computacional.

Aquest capítol proporciona una millora temporal en la detecció de soroll impulsu en imatges en color fent servir mètriques fuzzy. La velocitat computacional de la detecció del soroll s'ha comparat amb mètodes clàssics i recents i les experimentacions mostren que els avantatges del nostre algorisme poden ser molt adequat per aplicacions en temps real.

6 Millores en el Filtratge d'Imatges Mitjançant Tècniques de Peer Group

Article enviat a *Image and Vision Computing* com a “J.G. Camarena, V. Gregori, S. Morillas, A. Sapena, **Some Improvements for Image Filtering using Peer Group Techniques**”

Resum

El *peer group* d'un píxel d'una imatge es defineix com el conjunt dels seus veïns que són similars a ell d'acord a una distància o mesura de similitud adequada. Aquest concepte s'ha fet servir acuradament per a idear algorismes per a la detecció i eliminació de soroll impulsu en imatges en escala de grisos i en color. En aquest capítol, presentem una nova tècnica basada en el concepte de *peer group* enfocada a millorar el balanç entre l'eficiència computacional i la qualitat de filtratge davant els mètodes recents basats en *peer group*. Millorem l'eficiència computacional fent servir una modificació d'un treball recent que sols pot aplicar-se quan la mesura de similitud compleix la propietat de la desigualtat triangular. La millora de la qualitat del filtratge s'obté per un pas de refinament de la detecció de soroll. El filtre proposat actua en dos passos: (i) a una primera fase dividim la imatge en finestres disjunctes i efectuem una classificació ràpida dels píxels en tres classes, *no corruptes*, *no diagnosticats* i *corruptes provisionals*; a una segona fase refinem la classificació inicial analitzant els píxels *no diagnosticats* per a finalment tenir solament *no corruptes* i *corruptes*; (ii) per a concloure, sols els píxels diagnosticats com a *corruptes* es reemplacen, el que preserva les dades de la imatge no corruptes. Els estudis experimentals suggereixen que el mètode proposat es capaç de millorar els mètodes actuals de l'estat de l'art tant en qualitat del filtratge com en eficiència computacional.

6.1 Introducció

Els filtres basats en medianes vectorials [4, 84, 63, 71, 16] són mètodes extensament utilitzats per a la reducció de soroll en imatges en color i imatges multicanal a causa de dos motius principals:

1. Són capaços d'efectuar un filtratge robust ja que es basen en la teoria estadística robusta i
2. La correlació existent entre els canals de la imatge és necessari que es tinga en compte ja que les imatges es tracten com a camps vectorials.

D'altra banda, en aquests mètodes clàssics les operacions de filtratge s'aplica en cadascun dels píxels de la imatge sense tenir en compte si és o no sorollós, i llavors tendeixen a difuminar els vèrtexs i detalls de la imatge i per tant degradar la qualitat de la imatge filtrada. Amb tot això, a causa de la naturalesa no lineal, aquests mètodes tenen gran demanda computacional. S'han introduït recentment una sèrie de filtrats per substitució [62]-[12] per tal de sobreposar-se a aquests inconvenients. Aquests filtres segueixen un mètode de detecció i substitució per a que l'operació de filtratge s'aplique solament als píxels detectats com a sorollosos i, com a conseqüència, els píxels lliures de soroll es mantenen inalterats. Realment, quan la imatge està contaminada amb soroll impulsiu, els esquemes de substitució poden aplicar-se satisfactòriament a causa del seu bon rendiment i baixa complexitat computacional. Concretament, certs filtres vectorials estan basats en el concepte de *peer group* [21]-[12]. El *peer group* de un píxel d'una imatge és un conjunt format pels seus píxels veïns que són semblants a ell segons una adequada mesura de proximitat o distància. Els filtres introduïts per [21, 49] fan servir la distància entre el *peer group* del píxel analitzat i altres *peer groups* dins del seu veïnat com a regla de detecció. El treball en [38] proposa fer servir finestres de distinta grandària per a determinar la regió propera de cada píxel i llavors comprovar aquesta regió així com la forma d'aquesta. El mètode de [105] proposa que un píxel pot declarar-se com *lliure de soroll* si i sols si la cardinalitat del seu *peer group* és major que un llindar determinat. El mètode de [12] introdueix una modificació de [105] per efectuar un filtratge ràpid on el concepte de *peer group* es defineix amb un context de mètriques fuzzy. El filtre proposat a [89] separa el procés de detecció en dos fases anomenades: (i) detecció robusta i (ii) refinament. Els treballs en [75, 76] amplien el filtre definit en [49] al domini de la direccionalitat i el fan servir com a sub-filtre en un esquema híbrid.

En aquest capítol, proposem un nou mètode per a detecció de soroll impulsiu que fa servir *peer group* i funciona d'un mode ràpid i eficient. Com podrà veure's, l'algorisme proposat millora el balanç entre càrrega computacional i qualitat del filtratge respecte d'altres tècniques basades en *peer group*. Tot això s'obté per l'aplicació conjunta de conceptes introduïts en [89, 12] en un nou mètode de detecció de soroll. Particularment, el mètode combina la ràpida detecció de [12] amb la fase de refinament de [89]. L'objectiu principal del disseny actual és buscar un filtre centrant-se més en la qualitat del filtratge que en la càrrega computacional. Un cop el soroll s'ha detectat, s'aplica una tècnica adequada de filtratge o restauració únicament als píxels detectats com a corruptes. Proposem certs estudis experimentals per a mostrar el rendiment de l'esquema proposat.

En aquest estudi desenvolupem una nova tècnica de filtratge basant-nos únicament en un nou algorisme. Al llarg de tota la Tesi Doctoral hem fet servir la mètrica fuzzy ja exposada, però en aquest cas fem servir la mètrica L_2 , per tal de garantir que l'algorisme té un funcionament correcte independentment

de la mètrica utilitzada. Al final del capítol es presenta una variant de l'actual algorisme amb la mètrica fuzzy, per així no deixar d'analitzar la mètrica que tan bons resultats ens ha donat fins ara.

El capítol s'organitza com segueix. A la Secció 6.2 es detalla l'algorisme de filtratge proposat. L'estudi experimental i certes comparacions davant de filtres recents i coneguts es presenten a la Secció 6.3. La complexitat computacional s'analitza a la Secció 6.4. Posteriorment a la Secció 6.5 es mostra una implementació del filtre amb la famosa mètrica treballada al llarg de la Tesi Doctoral per a finalment a la Secció 6.6 presentar les conclusions.

6.2 Detecció i Eliminació de Píxels Corruptes

En aquest capítol representarem les imatges en color com una matriu bidimensional de dimensions $N_1 \times N_2$, a l'espai de color RGB on es fan servir 24 bits per emmagatzemar la informació de cada píxel, 8 per cada component de color. Així, cada píxel \mathbf{x}_i és un vector de tres components de valors enters dins del rang $[0..255]$. Per tal de descriure aquest nou mètode anomenat *Filtre Ràpid amb Peer Group Millorat* (IFPGF) anem a redefinir el concepte de *peer group* [105] d'un píxel \mathbf{x}_i inclòs a una finestra W $\mathcal{P}(\mathbf{x}_i, d) = \{\mathbf{x}_j \in W : \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq d\}$ on $d > 0$ i $\|\cdot\|_2$ representa la norma Euclídea. Noteu que $\forall d, \mathbf{x}_i \in \mathcal{P}(\mathbf{x}_i, d)$. Nosaltres hem pres en aquest capítol la coneguda mètrica Euclídea, encara que podria ser reemplaçada sota les mateixes condicions per qualsevol distància clàssica o mesura de similitud que acomplisca completament la desigualtat triangular, com per exemple les citades a [63], ja que la novetat del filtratge és el mètode i no la mètrica. No obstant, al final del capítol hem implementat una versió fuzzy de l'algorisme dissenyat, per seguir la línia d'investigació que dona fil conductor a la Tesi Doctoral.

El mètode proposat es divideix en un pas de detecció de soroll i un pas de filtratge de soroll. El pas de detecció de soroll s'efectua en dos fases. A la primera fase, la imatge es divideix en finestres quadrades disjunes. Per a cada finestra W de grandària $n \times n$: $n = 2w + 1$, $w = 1, 2, \dots$ centrada sobre \mathbf{x}_i apliquem el següent procediment prenent un valor fix de m en $\{1, \dots, n^2 - 1\}$:

SI $\#\mathcal{P}(\mathbf{x}_i, d) \geq (m + 1)$ ($\#$ representa la cardinalitat del conjunt)
 LLAVORS $\forall \mathbf{x}_j \in \mathcal{P}(\mathbf{x}_i, d)$, \mathbf{x}_j es declara com *no corrupte* i
 $\forall \mathbf{x}_k \in W - \mathcal{P}(\mathbf{x}_i, d)$, \mathbf{x}_k es declara com *no diagnosticat*
 SI NO \mathbf{x}_i es declara com *corrupte* provisionalment i
 $\forall \mathbf{x}_k \in W, \mathbf{k} \neq \mathbf{i}$, \mathbf{x}_k es declara com *no diagnosticat*

La idea del procés anterior resideix en que si $\mathcal{P}(\mathbf{x}_i, d)$ té almenys $m + 1$ membres llavors \mathbf{x}_i pot declarar-se com *no corrupte*, el que és la idea subjacent a [105]. Addicionalment a \mathbf{x}_i , tota la resta de píxels de $\mathcal{P}(\mathbf{x}_i, d)$ es declaren

com *no corruptes*. De fet, tenint en compte la desigualtat triangular, podem garantir que la distància entre dos píxels qualssevol de $\mathcal{P}(\mathbf{x}_i, d)$ serà menor o igual a $2d$. Llavors, amb la selecció d'un valor adequat de d , cada píxel de $\mathcal{P}(\mathbf{x}_i, d)$ pot declarar-se com *no corrupte* amb certa confiança. Aquest concepte, suggerit inicialment a [12], implica una reducció significativa dels càlculs respecte a [105] perquè els píxels diagnosticats addicionalment a \mathbf{x}_i no necessiten cap procés més.

Per a la segona fase del procés de detecció prenem un valor $m' \in \{1, \dots, m - 1\}$, i apliquem el següent procediment en cada píxel *no diagnosticat* centrat sobre \mathbf{x}_i en una finestra W de grandària $n \times n$:

SI $\#\mathcal{P}(\mathbf{x}_i, d) \geq (m + 1)$ o $\mathcal{P}(\mathbf{x}_i, d)$ conté m' píxels *no corruptes*
 LLAVORS $\forall \mathbf{x}_j \in \mathcal{P}(\mathbf{x}_i, d)$, \mathbf{x}_j es declara com *no corrupte*
 SI NO \mathbf{x}_i es declara com *corrupte*

En aquesta segona fase, afegim a la condició de $\mathcal{P}(\mathbf{x}_i, d)$ incloure m' veïns *no corruptes* que es pren del pas de refinament del mètode introduït per [89]. La idea sota aquesta condició es que si un píxel és similar a certs píxels *no corruptes* llavors pot considerar-se com *no corrupte*. Aquesta condició ens ajuda en la millora de l'exactitud de la detecció de soroll prop de vèrtexs i regions detallades on no és fàcil trobar m veïns propers. Això ens permetrà millorar la qualitat de la imatge de sortida del mètode de filtratge. Al final d'aquesta segona fase del procés de detecció, tot els píxels de la imatge han estat classificats bé com píxels *corruptes*, bé com píxels *no corruptes*. Noteu que certs píxels que podien haver estat declarats com *corruptes* provisionalment a la fase primera, poden diagnosticar-se en aquesta segona com *no corruptes*.

Finalment, el soroll detectat s'elimina aplicant una tècnica de filtratge en cada píxel detectat com *corrupte*. Ací farem servir el *Filtre de Mitjana Aritmètica* (AMF) [84] sobre els píxels veïns *no corruptes* en la finestra de filtratge, (seguint a [105, 12]), a causa de la simplicitat computacional, encara que, altres tècniques de filtratge vectorials com les introduïdes a [84, 63, 71, 16] poden aplicar-se. En cas que la finestra no continga cap píxel *no corrupte* s'ha de fer més gran fins que continga almenys un píxel *no corrupte* per a poder calcular la mitjana. L'algorisme 6 resumeix el mètode proposat.

6.3 Estudi Experimental

Dins d'aquesta Secció, s'han utilitzat distintes imatges per avaluar el comportament del filtre proposat. Per a les experimentacions, hem fet servir les imatges Caps, Flower, Girl i Statue de la Figura 6.1, preses de la base de dades d'imatges de test de KODAK [54]. Particularment, un detall menut de cada imatge s'ha fet servir per tal de apreciar millor el rendiment davant

Algorisme 6: Filtre Ràpid amb Peer Groups Millorat (*IFPGF*)

```

1 Dividim la imatge en finestres disjuntives  $W$  de grandària  $(2n + 1) \times (2n + 1)$ 
  centrades a  $\mathbf{x}_i$ , respectivament;
2 PerATot  $W$  Fer
3   calcular  $\mathcal{P}(\mathbf{x}_i, d)$ ;
4   Si  $\#\mathcal{P}(\mathbf{x}_i, d) \geq (m + 1)$  Llavors
5      $\forall \mathbf{x}_j \in \mathcal{P}(\mathbf{x}_i, d)$ ,  $\mathbf{x}_j$  es declara com no corrupte ;
6      $\forall \mathbf{x}_k \in W - \mathcal{P}(\mathbf{x}_i, d)$ ,  $\mathbf{x}_k$  es declara com no diagnosticat ;
7   Si no
8      $\mathbf{x}_i$  es declara com corrupte provisionalment ;
9      $\forall \mathbf{x}_k \in W, k \neq i$ ,  $\mathbf{x}_k$  es declara com no diagnosticat;
10 PerATot píxel no diagnosticat  $\mathbf{x}_i$  Fer
11   calcular  $\mathcal{P}(\mathbf{x}_i, d)$ ;
12   Si  $\#\mathcal{P}(\mathbf{x}_i, d) \geq (m + 1)$  O  $\mathcal{P}(\mathbf{x}_i, d)$  conté  $m'$  píxels no corruptes Llavors
13      $\forall \mathbf{x}_j \in \mathcal{P}(\mathbf{x}_i, d)$ ,  $\mathbf{x}_j$  es declara com no corrupte ;
14   Si no
15      $\mathbf{x}_i$  es declara com corrupte ;
16 PerATot píxel corrupte  $\mathbf{x}_i$  centrat en la finestra de filtratge  $W$  Fer
17   Si  $W$  no conté cap píxel no corrupte Llavors
18     Repetir
19       Augmentar el tamany de  $W$ ;
20     Fins Que  $W$  conté almenys un píxel no corrupte ;
21   Substituir  $\mathbf{x}_i$  amb la sortida del filtre AMF aplicat als píxels no
    corruptes de  $W$ ;

```

distints ajusts de paràmetres i mètodes de filtratge. Aquestes imatges s'han contaminat amb soroll impulsiu seguint els models de soroll amb valors aleatoris i soroll amb valors fixos de [84, 89]. Seguidament, representarem per p la intensitat de soroll podent-la considerar com la probabilitat d'aparició de soroll. El filtre proposat es compara davant el filtre clàssic *Vector Median Filter* [4], el recent mètode adaptatiu *Fast Impulsive Vector Filter* (FIVF) [88] i *New Adaptive Vector Median Filter* (NAVMF) [90], i els recents mètodes basats en tècniques de *peer group*, *Peer Group Filter* (PGF) [105], *Fast Peer Group Filter* (FPGF) [12] i *Iterative Peer Group Filter* (IPGF) [89]. Els rendiments dels filtres s'ha calculat mitjançant les mesures de bondat objectives descrites a [84, 63, 71] *Error Mitjà Absolut* (MAE), *Pics de Senyal en Ratios de Soroll* (PSNR) i *Diferència de Color Normalitzat* (NCD) en tal de tenir en compte els aspectes de conservació del senyal, eliminació de soroll i el criteri d'error perceptual humà, respectivament.



(a)

(b)



(c)

(d)



(e)

(f)

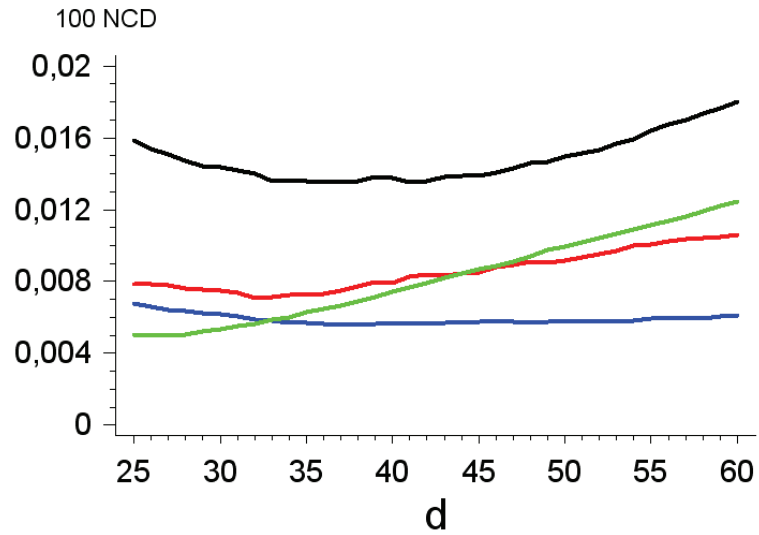


(g)

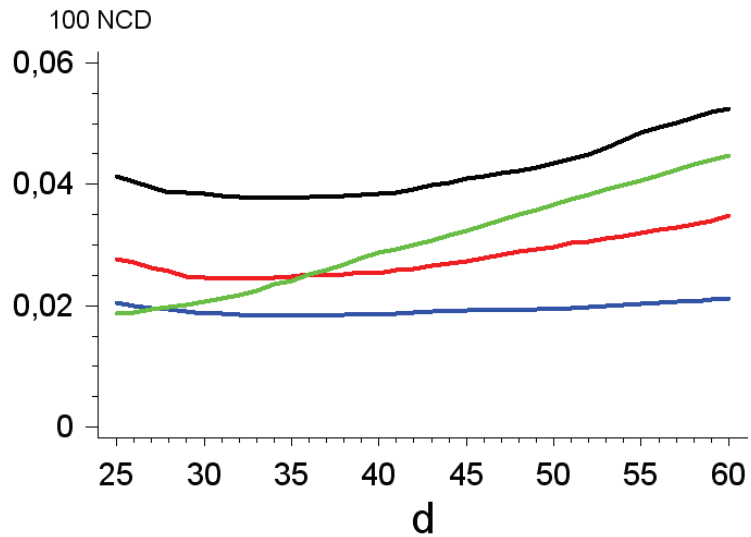
(h)

Figura 6.1. Imatges per a test de la base de dades KODAK [54]: (a-b) Caps i detall respectiu, (c-d) Girl i detall respectiu, (e-f) Flower i detall respectiu, (e-h) Statue i detall respectiu.

Inicialment, en tal d'ajustar adequadament el valor del paràmetre d hem fet un estudi del rendiment del filtre en funció del paràmetre d . Alguns dels resultats obtinguts es veuen a les Figures 6.2 i 6.3. Hem observat que no existeix un valor òptim genèric de d , ja que aquest òptim difereix per a cada imatge i per a cada intensitat de soroll. Tot i això, hem observat que un valor robust capaç d'obtenir resultats quasi-òptims està al voltant de $d = 35$, i per tant, proposem aquest valor. També s'observa com la imatge Statue (línia negra) sempre està per damunt en totes les gràfiques, ja que aquesta imatge és la que conté més detall. Per contra la imatge Caps (línia blava) sempre està per davall, ja que és la més homogènia i, per tant, la més senzilla de filtrar. El comportament de totes elles és el mateix sota diferents condicions d'intensitat de soroll com del paràmetre d .

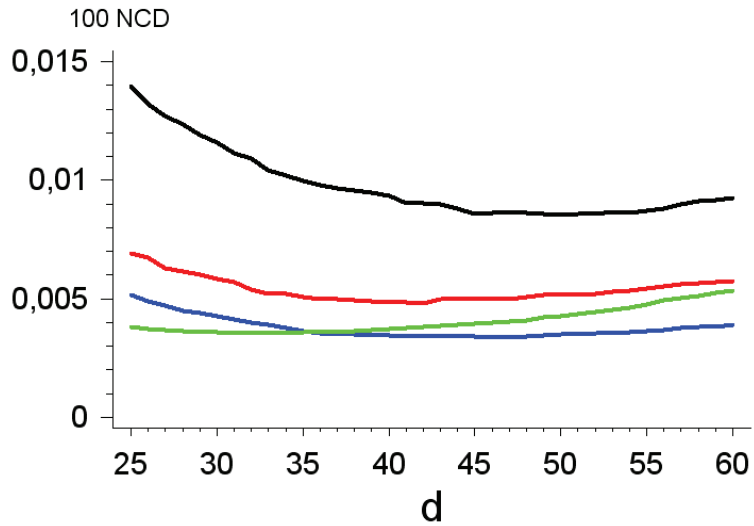


(a)

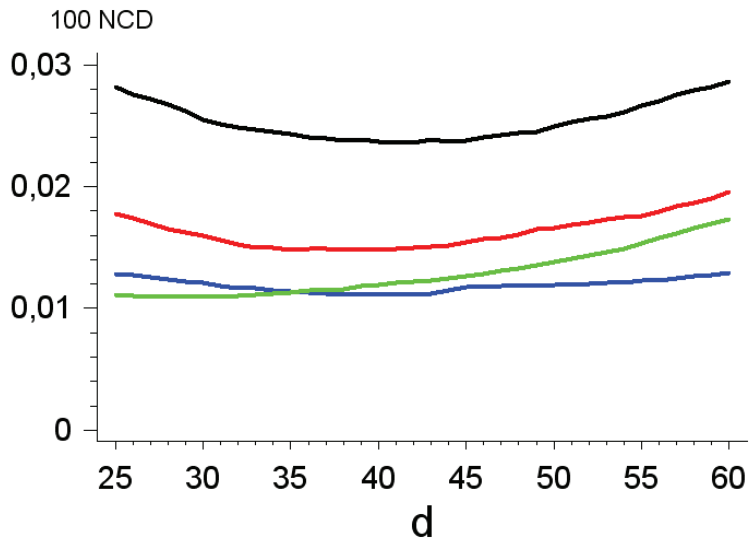


(b)

Figura 6.2. Rendiment del filtre IFPGF en termes de NCD en funció del paràmetre d per a diferents imatges en distintes intensitats de soroll impulsiu: (a) 10% de soroll i (b) 30% de soroll. En color Blau es representa la imatge Caps, en Roig es representa la imatge Flower, en color Verd es representa la imatge Girl i en color Negre es representa la imatge Statue.



(a)



(b)

Figura 6.3. Rendiment del filtre IFPGF en termes de NCD en funció del paràmetre d per a diferents imatges en distintes intensitats de soroll uniforme: (a) 10% de soroll i (b) 30% de soroll. En color Blau es representa la imatge Caps, en Roig es representa la imatge Flower, en color Verd es representa la imatge Girl i en color Negre es representa la imatge Statue.

Amb respecte als paràmetres m i m' hem observat que per a finestres de filtratge 3×3 els millors resultats s'obtenen per a $m = 3, m' = 1$ per a soroll de valor fix (excepte quan la intensitat de soroll és baixa ($< 10\%$) que el valor més apropiat és $m = 2, m' = 1$). Això és perquè en els casos de soroll de valor fix de vegades poden aparèixer blocs de soroll similar (píxels sorollosos semblants entre si), que s'eliminen millor per a valors més grans de m . Les Figures 6.4 i 6.5 mostren dos exemples del funcionament del procés de detecció de soroll on pot veure's que la detecció de soroll és robusta i encertada.

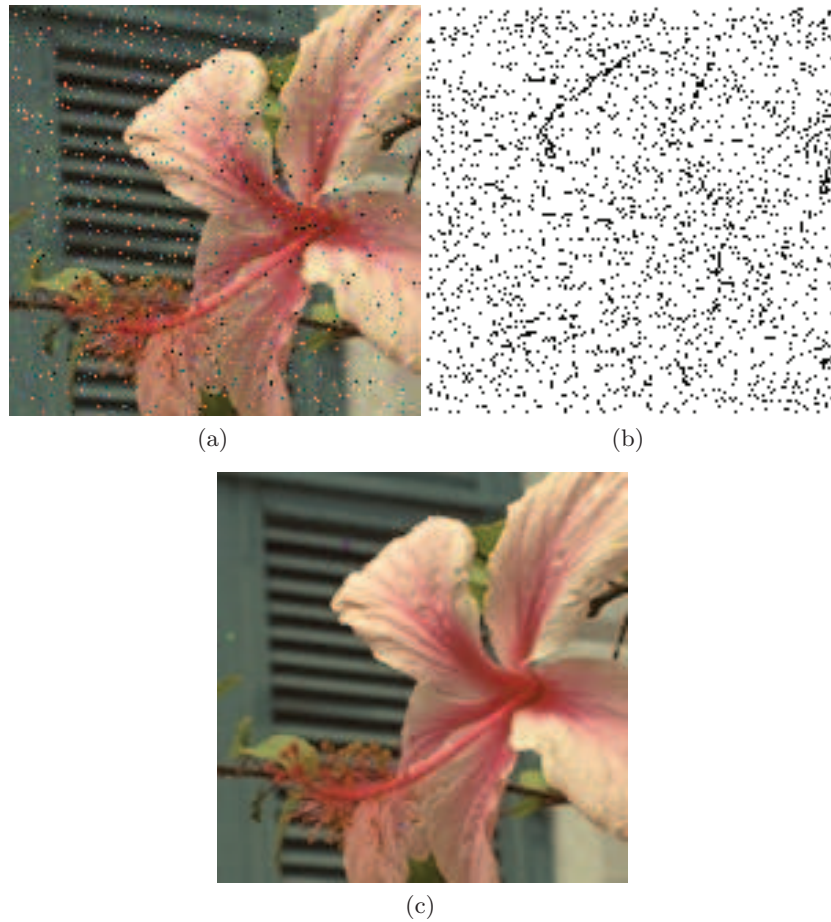


Figura 6.4. Exemple de rendiment del filtre IFPGF: (a) imatge Flower amb 10% de soroll impulsiu de valor fix, (b) soroll detectat (píxels negres), (c) sortida del filtre IFPGF.

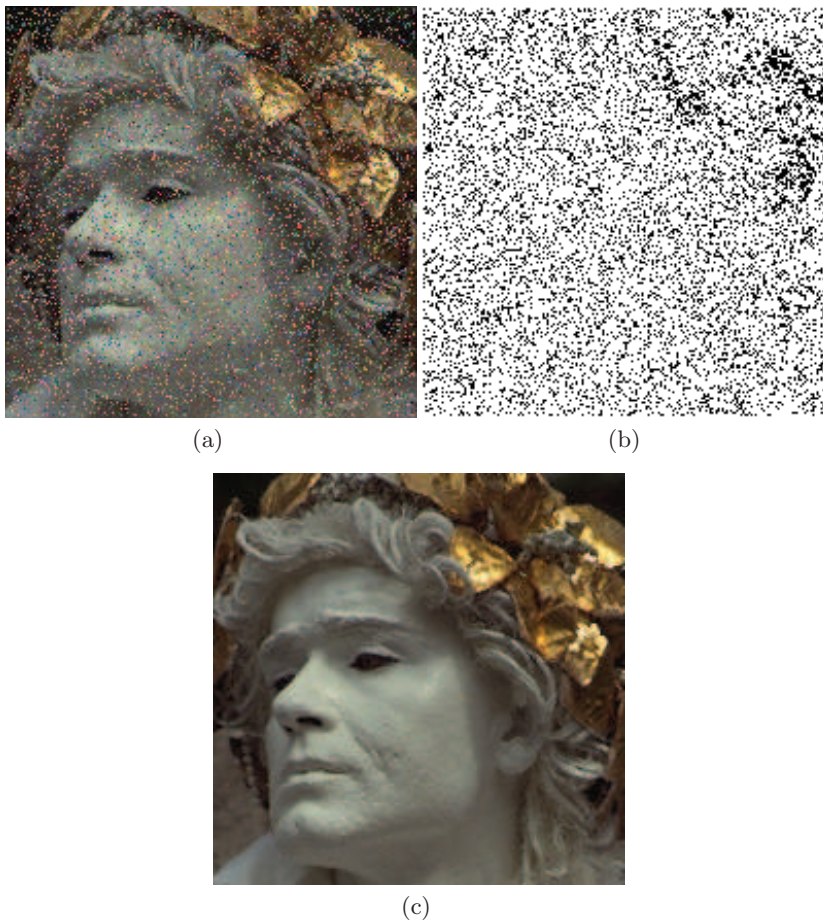


Figura 6.5. Exemple de rendiment del filtre IFPGF: (a) imatge Statue amb 20% de soroll impulsiu de valor aleatori, (b) soroll detectat (píxels negres), (c) sortida del filtre IFPGF.

També hem mesurat a l'estudi els ratios de píxels corruptes i no corruptes detectats de manera correcta i incorrecta amb els valors establerts dels paràmetres. Aquestes ratios descrits responen a:

Encerts Positius	Píxels corruptes detectats com corruptes	SC
Encerts Negatius	Píxels no corruptes detectats com no corruptes	NSC
Errades Positives	Píxels corruptes detectats com no corruptes	SI
Errades Negatives	Píxels no corruptes detectats com corruptes	NSI

Els valors obtinguts es poden consultar a les Taules 6.1-6.2. D'aquests resultats pot observar-se que la precisió del mètode és elevada, ja que els percentatges de píxels sorollosos i no sorollosos detectats incorrectament (SI *Sorollosos Incorrectes* i NSI *No Sorollosos Incorrectes*, respectivament) són baixos. També, el percentatge de soroll detectat correctament (SC *Sorollosos Correctes*) és molt proper al percentatge de contaminació de soroll, cosa que indica que el rendiment del filtre és robust. Tot i això, pot veure's que certes vegades el soroll detectat és sensiblement inferior al percentatge real de soroll. Això és perquè certes vegades els píxels sorollosos són molt semblants al fons de la imatge al lloc on apareix el soroll i en aquests casos el soroll no es detecta encertadament.

Taula 6.1. Estadístiques (percentatges) de la detecció efectuada de píxels sorollosos i lliures de soroll: No Sorollosos Correctes (NSC), No Sorollosos Incorrectes (NSI), Sorollosos Correctes (SC) i Sorollosos Incorrectes (SI), per a distintes imatges corruptes amb distintes intensitats de soroll impulsiu de valor fix.

p	0.05				0.10			
Imatge	NSC	NSI	SC	SI	NSC	NSI	SC	SI
Caps	94.26	0.76	4.31	0.67	89.52	0.85	8.18	1.45
Flower	93.82	0.98	4.94	0.26	89.12	1.20	9.07	0.62
Girl	94.90	0.09	4.71	0.30	89.96	0.22	9.30	0.53
Statue	93.04	1.87	4.79	0.31	88.07	1.96	9.31	0.66

p	0.20				0.30			
Imatge	NSC	NSI	SC	SI	NSC	NSI	SC	SI
Caps	79.63	0.98	16.29	3.09	68.85	2.05	24.82	4.28
Flower	76.48	3.53	18.77	1.21	65.72	4.07	28.30	1.91
Girl	79.61	0.32	18.86	1.21	69.55	0.55	27.84	2.06
Statue	77.81	2.24	18.13	1.81	66.52	3.95	27.72	1.81

El resultat experimental (obtinguts en un Pentium IV, 3.4 GHz, 512 MB RAM) a les Taules 6.3-6.10 mostren la comparativa en termes de mesures de qualitat objectives. Pot comprovar-se que el mètode proposat millora totes les altres tècniques analitzades exceptuant el IPGF. Recordeu que el filtre IPGF (*Iterative Peer Group Filter* és un filtre iteratiu, que repeteix els procés de filtratge un nombre indeterminat de vegades fins arribar a una qualitat o lliurar desitjats). Evidentment, l'avantatge del mètode proposat davant del filtre IPGF és el rendiment computacional que pot veure's a les Figures de comparativa de rendiment (6.6 a 6.9) i s'estudia a la secció següent.

Taula 6.2. Estadístiques (percentatges) de la detecció efectuada de píxels sorollosos i lliures de soroll: No Sorollosos Correctes (NSC), No Sorollosos Incorrectes (NSI), Sorollosos Correctes (SC) i Sorollosos Incorrectes (SI), per a distintes imatges corruptes amb distintes intensitats de soroll impulsiu de valor aleatori.

p	0.05				0.10			
Imatge	NSC	NSI	SC	SI	NSC	NSI	SC	SI
Caps	93.99	0.84	5.07	0.09	89.19	0.82	9.88	0.11
Flower	93.87	1.03	4.98	0.12	88.76	1.26	9.75	0.23
Girl	94.92	0.09	4.90	0.09	89.82	0.10	9.89	0.19
Statue	93.39	1.81	4.66	0.13	88.40	1.87	9.54	0.19

p	0.20				0.30			
Imatge	NSC	NSI	SC	SI	NSC	NSI	SC	SI
Caps	79.09	0.99	19.66	0.26	68.67	1.33	29.62	0.38
Flower	78.73	1.45	19.39	0.42	68.01	2.07	29.32	0.61
Girl	79.89	0.16	19.61	0.34	69.67	0.30	29.49	0.53
Statue	77.87	2.19	19.56	0.37	67.37	2.53	29.51	0.59

Com es pot veure a les Figures 6.6- (6.9, a la subfigura (a) el guany respecte al filtre FMPGF (exposat al Capítol 3) en quant a termes de qualitat (línia sòlida) és considerable, estant sempre per davall en totes les intensitats de soroll així com en totes les imatges. El guany en quant a temps computacional (línia puntejada) s'observa que està al voltant del doble de ràpid per terme mitjà.

En la subfigura (b) d'aquestes imatges, s'observa que el filtre iteratiu IPGF, llevat de la imatge Girl, obté una millor qualitat de filtratge. Això és perquè a que aquest filtre efectua un nombre indeterminat d'iteracions fins que no pot millorar la qualitat del filtratge. Això provoca que aquest filtre siga computacionalment molt costós, i com s'observa a les figures està al voltant de 13 vegades més lent que el filtre que proposem.

Pot observar-se que la tècnica proposada millora el balanç entre la eficiència computacional¹ i la qualitat del filtratge, el que fa que aquest mètode siga molt pràctic per a aplicacions reals. Representem el temps computacional necessari per cada mètode com al ratio respecte al temps per al filtre de referència VMF per una millor comparativa visual. També, pot veure's que el ratio respecte al temps computacional de filtre IFPGF és significativament menor que el PGF. Per a una comparativa visual, a les Figures 6.10-6.13, s'observa que el mètode proposat genera imatges visualment correctes ja que tant els detalls de les imatges com els vèrtexs es preserven adequadament

¹ Temps computacional mesurat en implementacions optimitzades del filtres

Taula 6.3. Resultats experimentals dels filtres comparats amb la imatge Caps corrupta amb distintes densitats de soroll impulsiu de valor fix p .

p	0.05			0.10		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	2.54	21.76	2.96	4.84	18.97	5.57
VMF	2.65	32.11	1.83	2.80	31.66	1.92
FIVF	0.31	37.95	0.30	0.63	35.22	0.54
NAVMF	0.25	39.87	0.25	0.51	36.33	0.49
PGF	0.41	37.43	0.33	0.66	34.63	0.59
FPGF	0.39	37.26	0.34	0.66	34.16	0.62
IPGF	0.23	39.32	0.25	0.41	37.79	0.43
IFPGF	0.35	37.88	0.29	0.64	34.18	0.57
p	0.20			0.30		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	9.97	15.81	11.42	14.67	14.15	17.02
VMF	3.05	30.94	2.08	3.38	29.89	2.30
FIVF	1.21	32.41	1.02	1.85	29.77	1.62
NAVMF	1.16	32.57	0.99	1.85	29.99	1.58
PGF	1.28	31.24	1.23	2.17	29.14	1.96
FPGF	1.54	29.59	1.40	3.31	26.31	2.33
IPGF	0.86	34.91	0.85	1.39	32.61	1.31
IFPGF	1.34	31.72	1.11	2.12	28.50	1.83

Taula 6.4. Resultats experimentals dels filtres comparats amb la imatge Caps corrupta amb distintes densitats de soroll impulsiu de valor aleatori p .

p	0.05			0.10		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	4.29	20.75	3.72	8.31	17.86	7.22
VMF	2.74	31.68	1.88	2.96	31.07	2.00
FIVF	0.32	38.32	0.21	0.66	35.11	0.40
NAVMF	0.30	38.98	0.19	0.55	36.45	0.36
PGF	0.44	37.59	0.26	0.63	36.25	0.40
FPGF	0.41	38.05	0.25	0.63	36.35	0.39
IPGF	0.27	39.23	0.18	0.52	36.66	0.34
IFPGF	0.38	38.50	0.23	0.59	36.78	0.36
p	0.20			0.30		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	16.77	14.79	14.55	25.28	13.00	21.97
VMF	3.54	29.12	2.37	4.76	25.71	3.38
FIVF	1.34	31.71	0.82	2.16	28.65	1.37
NAVMF	1.28	32.16	0.81	2.27	28.31	1.49
PGF	1.20	33.42	0.77	1.92	30.56	1.23
FPGF	1.35	32.62	0.83	2.49	28.69	1.41
IPGF	1.15	32.62	0.78	2.31	27.78	1.67
IFPGF	1.16	33.44	0.72	1.88	30.61	1.14

Taula 6.5. Resultats experimentals dels filtres comparats amb la imatge Flower corrupta amb distintes densitats de soroll impulsiu de valor fix p .

p	0.05			0.10		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	2.55	22.26	3.93	4.86	19.44	7.16
VMF	4.37	29.47	2.29	4.60	29.15	2.42
FIVF	0.58	35.16	0.55	1.16	32.29	0.91
NAVMF	0.55	36.00	0.46	1.05	33.00	0.88
PGF	0.69	35.11	0.41	1.07	32.83	0.87
FPGF	0.67	35.56	0.44	1.13	32.47	0.91
IPGF	0.38	37.23	0.28	0.70	35.17	0.50
IFPGF	0.58	36.24	0.35	0.99	33.30	0.72
p	0.20			0.30		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	10.11	16.22	14.81	15.15	14.48	22.59
VMF	5.18	28.34	2.73	5.86	27.29	3.15
FIVF	2.19	29.79	1.52	3.38	27.10	2.44
NAVMF	2.13	30.09	1.50	3.43	27.53	2.34
PGF	2.05	29.36	1.97	3.70	27.10	2.81
FPGF	2.70	27.43	1.94	5.24	23.69	3.82
IPGF	1.45	32.79	0.92	2.48	29.88	1.58
IFPGF	2.29	29.52	1.51	3.56	26.99	2.47

Taula 6.6. Resultats experimentals dels filtres comparats amb la imatge Flower corrupta amb distintes densitats de soroll impulsiu de valor aleatori p .

p	0.05			0.10		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	3.77	21.80	3.52	7.58	18.71	6.80
VMF	4.47	29.37	2.34	5.03	28.34	2.64
FIVF	0.56	35.73	0.34	1.28	31.70	0.68
NAVMF	0.58	36.09	0.34	1.16	32.62	0.65
PGF	0.70	34.98	0.33	1.12	33.07	0.55
FPGF	0.70	35.28	0.34	1.17	32.83	0.58
IPGF	0.43	37.59	0.25	0.94	33.95	0.55
IFPGF	0.60	36.18	0.29	1.05	33.74	0.51
p	0.20			0.30		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	14.79	15.88	13.47	22.34	14.10	20.30
VMF	6.12	26.98	3.24	7.41	25.29	4.05
FIVF	2.42	29.30	1.29	3.70	27.03	2.01
NAVMF	2.33	29.66	1.24	3.73	27.35	1.99
PGF	1.91	31.11	1.00	3.00	28.79	1.66
FPGF	2.41	29.64	1.16	4.43	26.48	2.12
IPGF	1.94	30.82	1.12	3.36	27.64	2.01
IFPGF	1.85	31.45	0.93	2.94	29.21	1.48

Taula 6.7. Resultats experimentals dels filtres comparats amb la imatge Girl corrompuda amb distintes densitats de soroll impulsiu de valor fix p .

p	0.05			0.10		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	2.52	22.29	4.26	4.96	19.34	8.38
VMF	3.43	32.87	2.36	3.53	32.65	2.44
FIVF	0.35	38.75	0.58	0.68	36.92	0.75
NAVMF	0.31	40.23	0.46	0.58	37.93	0.73
PGF	0.30	40.78	0.41	0.56	37.24	0.86
FPGF	0.32	39.71	0.51	0.61	35.86	0.96
IPGF	0.24	41.46	0.36	0.43	39.85	0.52
IFPGF	0.27	41.48	0.34	0.53	39.06	0.64
p	0.20			0.30		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	10.12	16.24	17.17	14.99	14.52	25.51
VMF	3.76	32.18	2.60	4.01	31.20	2.83
FIVF	1.30	34.56	1.26	1.99	31.24	2.01
NAVMF	1.25	34.58	1.23	2.07	30.90	1.96
PGF	1.30	31.48	2.34	1.97	31.45	3.28
FPGF	1.50	29.80	2.07	3.05	25.36	3.85
IPGF	0.87	37.66	0.78	1.41	34.76	1.08
IFPGF	1.10	35.22	1.42	1.85	30.91	2.51

Taula 6.8. Resultats experimentals dels filtres comparats amb la imatge Girl corrompuda amb distintes densitats de soroll impulsiu de valor aleatori p .

p	0.05			0.10		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	3.80	21.77	3.94	7.60	18.78	7.87
VMF	3.47	32.79	2.39	3.59	32.51	2.48
FIVF	0.32	40.53	0.23	0.68	37.43	0.47
NAVVMF	0.31	41.21	0.23	0.60	38.53	0.43
PGF	0.28	41.69	0.20	0.52	39.71	0.39
FPGF	0.30	41.13	0.21	0.54	39.37	0.38
IPGF	0.28	41.65	0.21	0.53	39.31	0.39
IFPGF	0.27	42.24	0.19	0.50	40.17	0.36
p	0.20			0.30		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	15.08	15.79	15.56	22.62	14.05	23.42
VMF	3.96	31.40	2.77	4.71	28.82	3.38
FIVF	1.29	34.74	0.90	1.99	32.04	1.42
NAVVMF	1.28	34.72	0.89	2.11	31.51	1.48
PGF	1.06	36.35	0.82	1.71	33.73	1.39
FPGF	1.12	36.22	0.76	1.94	33.38	1.27
IPGF	1.11	35.69	0.82	1.94	31.41	1.48
IFPGF	1.01	37.06	0.72	1.58	34.73	1.13

Taula 6.9. Resultats experimentals dels filtres comparats amb la imatge Statue corrupta amb distintes densitats de soroll impulsiu de valor fix p .

p	0.05			0.10		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	2.53	22.30	4.45	5.03	19.21	8.67
VMF	5.39	26.92	3.72	5.58	26.72	3.84
FIVF	1.22	29.17	0.99	1.94	27.83	1.49
NAVMF	0.90	30.80	0.75	1.45	29.31	1.27
PGF	1.58	27.76	0.98	1.91	27.57	1.48
FPGF	1.46	28.26	0.99	1.98	27.39	1.62
IPGF	0.57	33.38	0.42	1.10	30.97	0.79
IFPGF	1.36	28.47	0.86	1.86	27.45	1.36
p	0.20			0.30		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	10.04	16.24	17.49	14.82	14.54	25.89
VMF	6.04	26.22	4.14	6.45	25.82	4.46
FIVF	2.97	26.59	2.38	4.02	25.24	3.51
NAVMF	2.65	27.35	2.18	3.84	25.85	3.26
PGF	2.87	26.12	2.92	4.80	23.96	4.34
FPGF	3.70	24.57	3.17	6.10	22.08	5.40
IPGF	2.05	28.98	1.50	3.15	27.43	2.38
IFPGF	3.07	25.22	2.87	4.70	23.41	3.77

Taula 6.10. Resultats experimentals dels filtres comparats amb la imatge Statue corrupta amb distintes densitats de soroll impulsiu de valor aleatori p .

p	0.05			0.10		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	3.59	21.99	3.71	7.39	18.82	7.53
VMF	5.49	26.83	3.80	5.82	26.44	4.01
FIVF	1.22	29.17	0.70	1.97	27.78	1.19
NAVFMF	0.92	30.64	0.55	1.49	29.30	0.94
PGF	1.52	28.02	0.81	1.88	27.82	1.09
FPGF	1.40	28.61	0.77	1.89	27.85	1.09
IPGF	0.67	32.66	0.43	1.25	30.62	0.83
IFPGF	1.29	28.84	0.69	1.75	28.10	1.00
p	0.20			0.30		
Filtre	MAE	PSNR	NCD	MAE	PSNR	NCD
Sorollosa	15.10	15.75	15.52	22.67	13.99	23.29
VMF	6.63	25.63	4.54	7.81	24.24	5.35
FIVF	3.04	26.69	1.93	4.31	25.13	2.86
NAVFMF	2.70	27.55	1.74	4.26	25.41	2.82
PGF	2.73	27.02	1.71	3.88	25.53	2.57
FPGF	3.35	25.81	1.99	5.55	23.41	3.38
IPGF	2.43	28.18	1.62	4.00	25.77	2.84
IFPGF	2.73	26.74	1.64	3.92	25.27	2.43

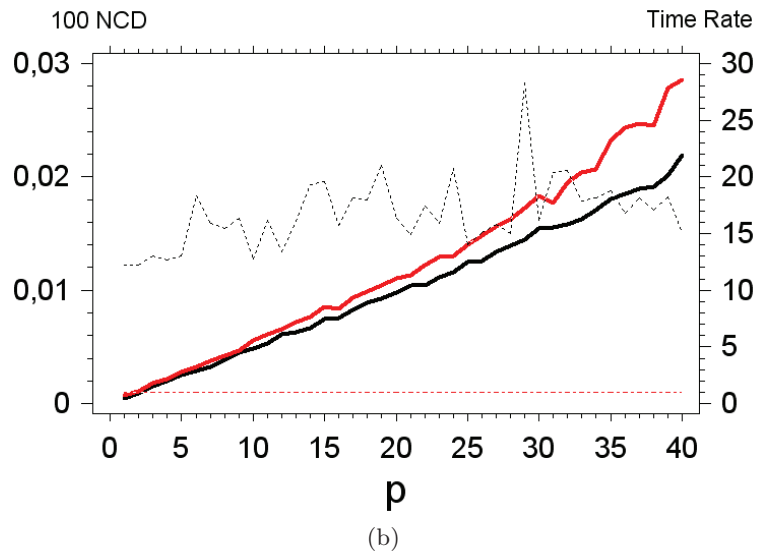
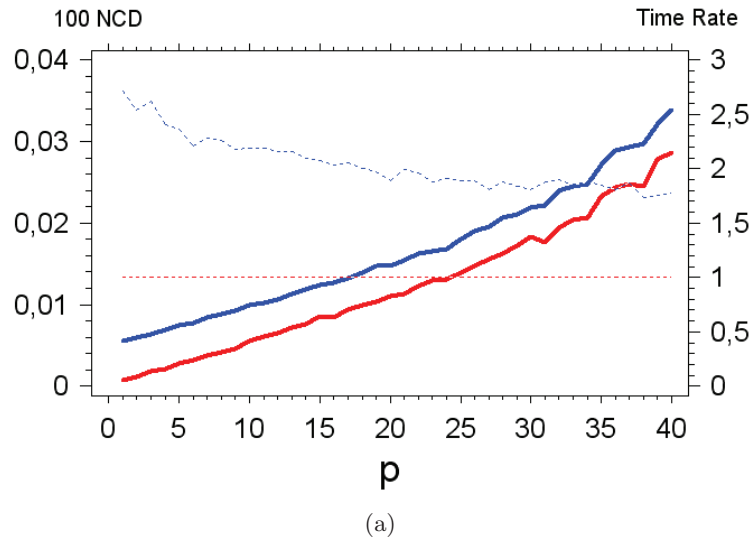
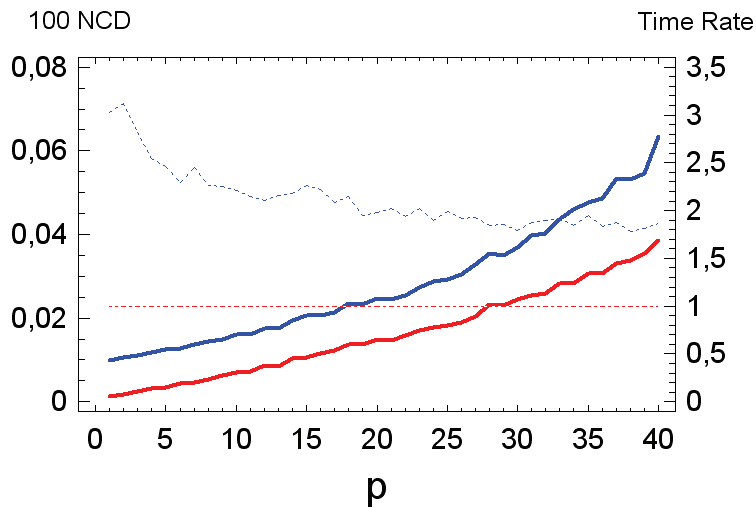
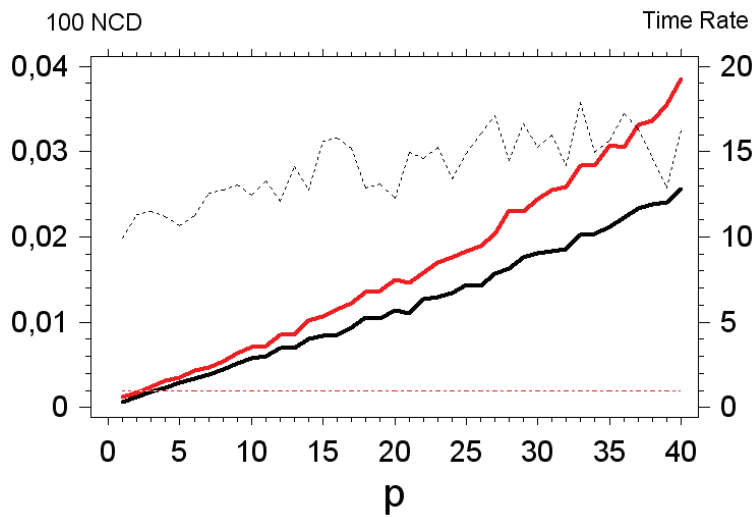


Figura 6.6. Comparativa de rendiments en termes de NCD (línies sòlides) amb el temps computacional (línies puntejades) per a la imatge Caps corrupta amb distintes probabilitats p d'aparició de soroll impulsiu de valor fix (a) i de valor aleatori (b). El temps computacional es representa com el ratio respecte al filtre proposat IFPGF. El color Blau correspon al filtre FPGF, el Negre al filtre IPGF i el Roig al filtre IFPGF.

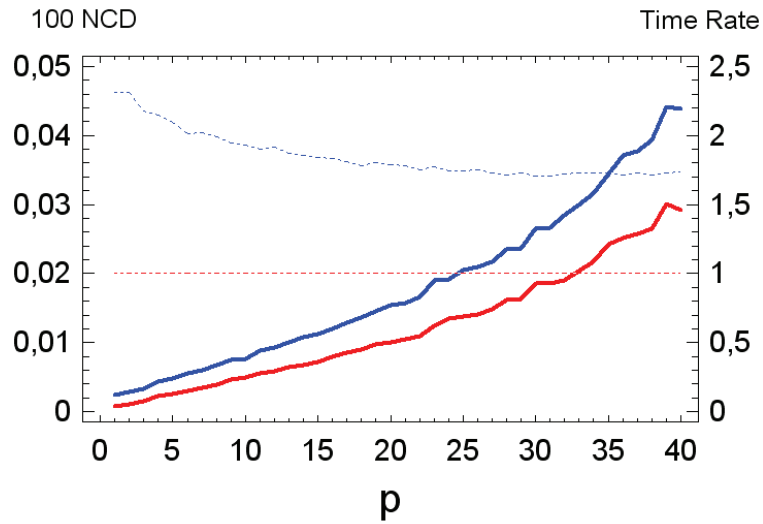


(a)

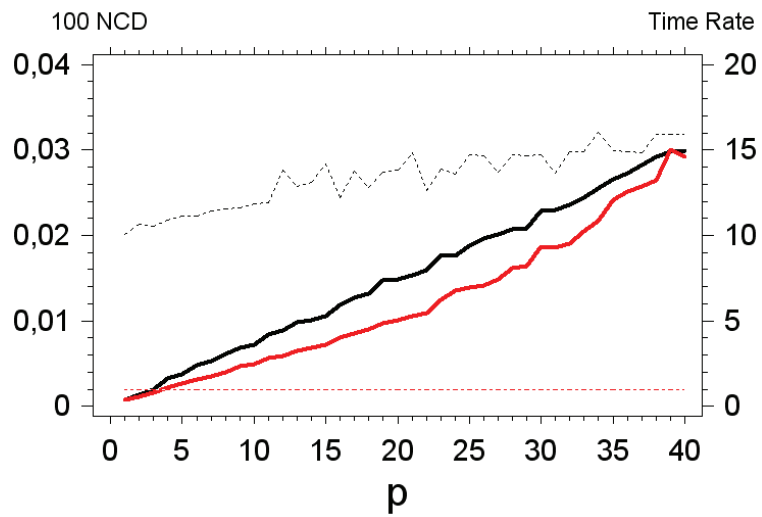


(b)

Figura 6.7. Comparativa de rendiments en termes de NCD (línies sòlides) amb el temps computacional (línies puntejades) per a la imatge Flower corrupta amb distintes probabilitats p d'aparició de soroll impulsu de valor fix (a) i de valor aleatori (b). El temps computacional es representa com el ratio respecte al filtre proposat IFPGF. El color Blau correspon al filtre FPGF, el Negre al filtre IPGF i el Roig al filtre IFPGF.



(a)



(b)

Figura 6.8. Comparativa de rendiments en termes de NCD (línies sòlides) amb el temps computacional (línies puntejades) per a la imatge Girl corrompuda amb distintes probabilitats p d'aparició de soroll impulsiu de valor fix (a) i de valor aleatori (b). El temps computacional es representa com el ratio respecte al filtre proposat IFPGF. El color Blau correspon al filtre FPGF, el Negre al filtre IPGF i el Roig al filtre IFPGF.

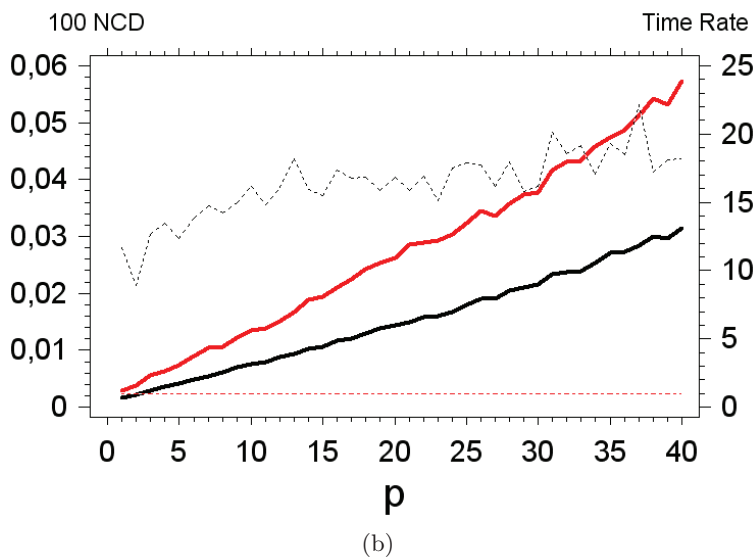
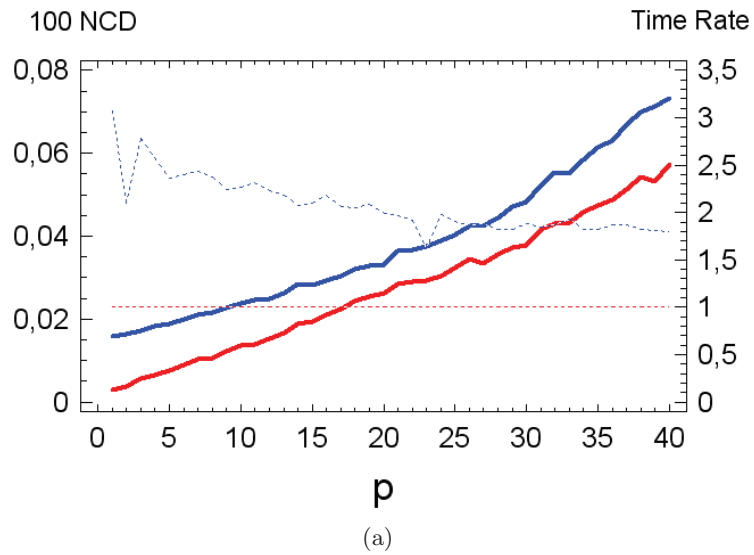


Figura 6.9. Comparativa de rendiments en termes de NCD (línies sòlides) amb el temps computacional (línies puntejades) per a la imatge Statue corrupta amb distintes probabilitats p d'aparició de soroll impulsiu de valor fix (a) i de valor aleatori (b). El temps computacional es representa com el ratio respecte al filtre proposat IFPGF. El color Blau correspon al filtre FPGF, el Negre al filtre IPGF i el Roig al filtre IFPGF.

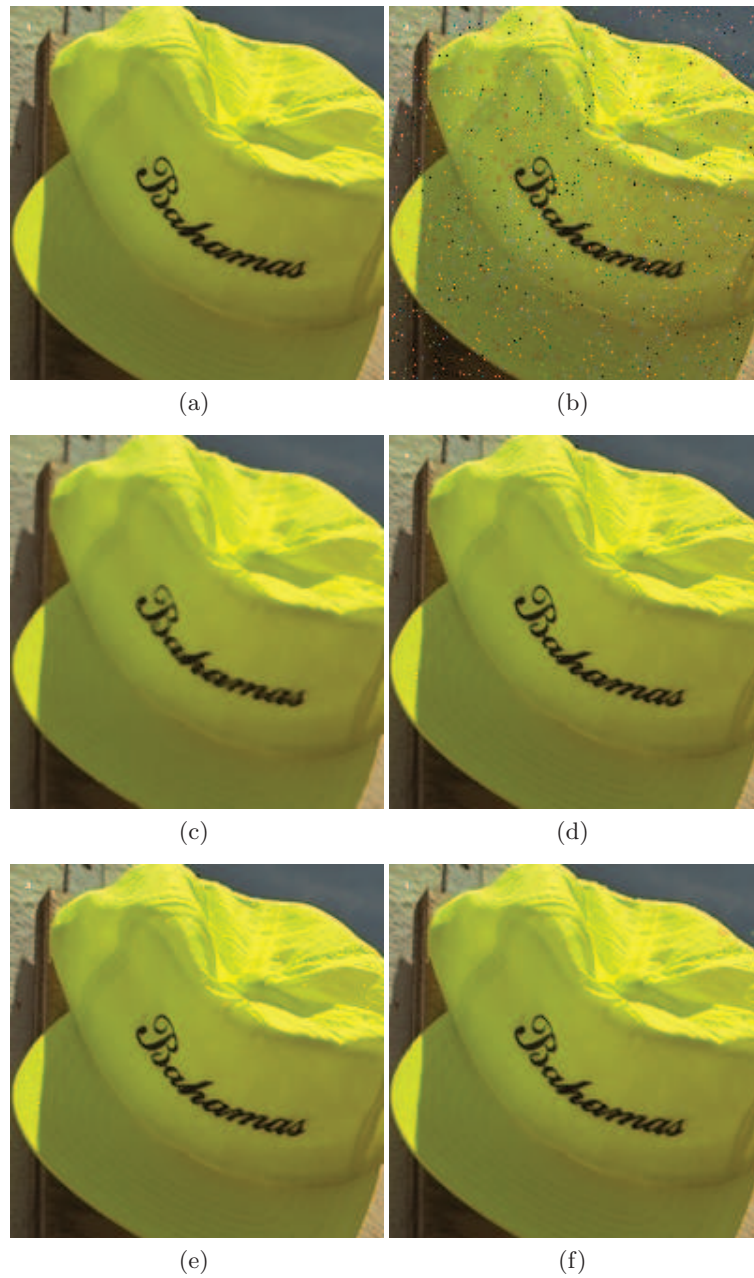


Figura 6.10. Comparativa de rendiment de filtres: (a) Imatge Caps original, (b) Imatge contaminada amb un 5% de soroll impulsiu de valor fix, i sortides de (c) VMF, (d) NAVMF, (e) IPGF, i (f) IFPGF.

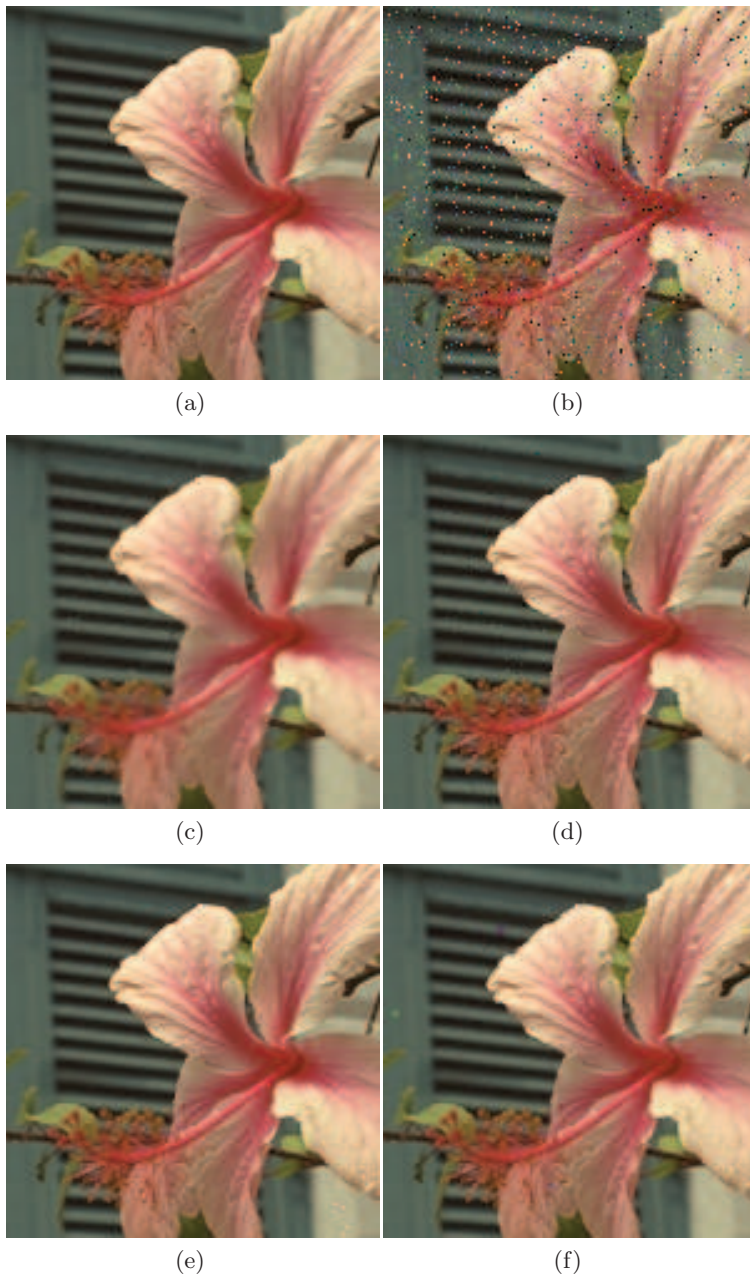


Figura 6.11. Comparativa de rendiment de filtres: (a) Imatge Flower original, (b) Imatge contaminada amb un 10% de soroll impulsiu de valor aleatori, i sortides de (c) VMF, (d) NAVMF, (e) IPGF, i (f) IFPGF.

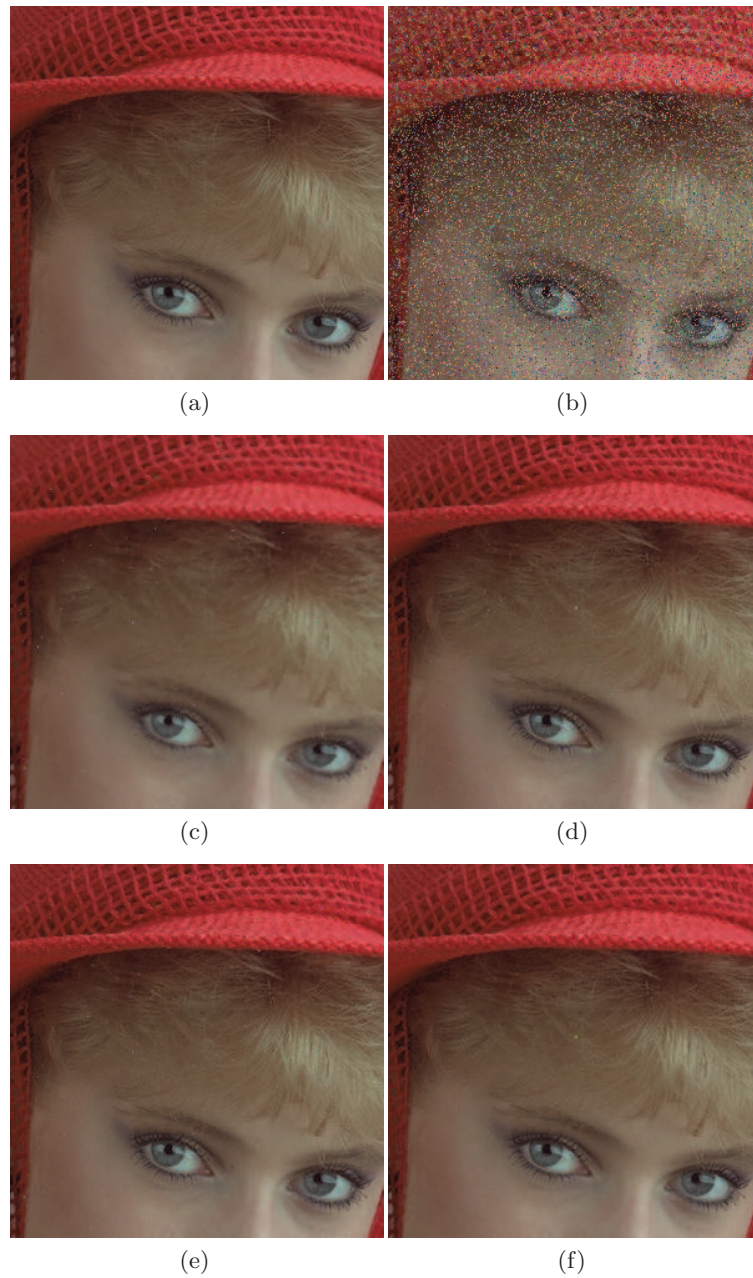


Figura 6.12. Comparativa de rendiment de filtres: (a) Imatge Girl original, (b) Imatge contaminada amb un 20% de soroll impulsiu de valor aleatori, i sortides de (c) VMF, (d) NAVMF, (e) IPGF, i (f) IFPGF.

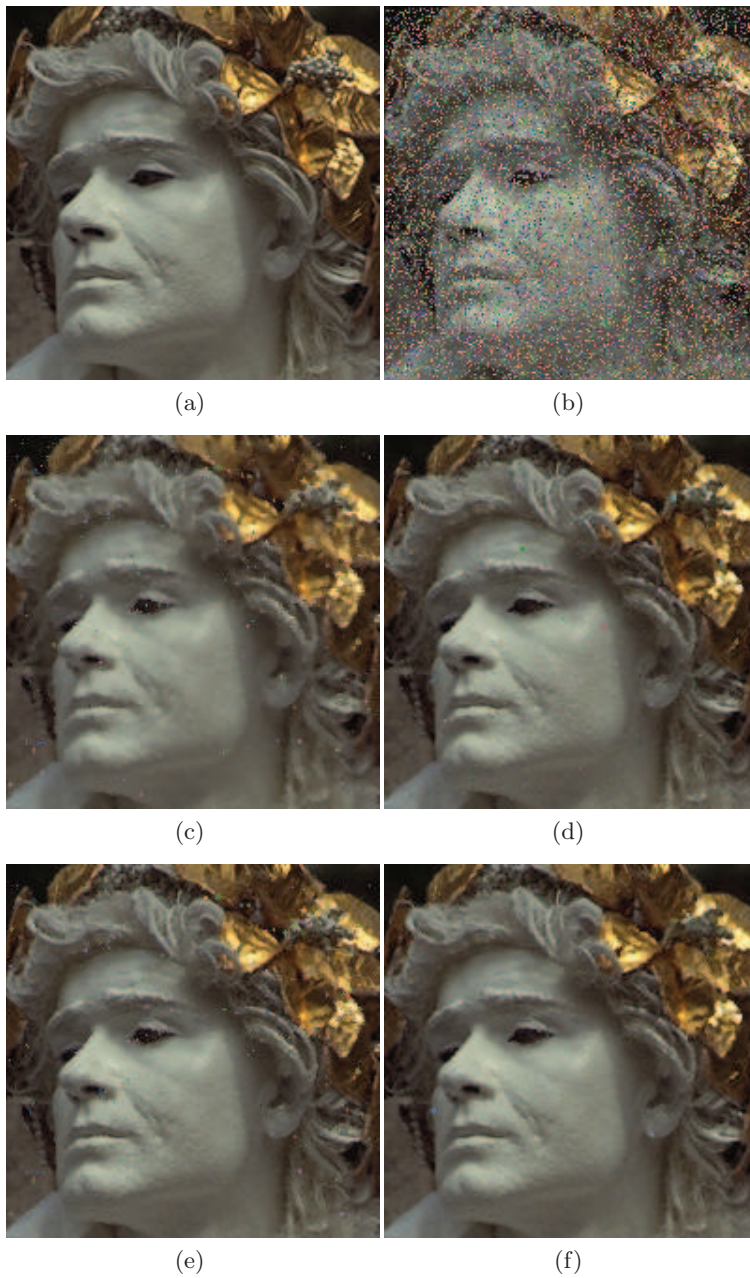


Figura 6.13. Comparativa de rendiment de filtres: (a) Imatge Statue original, (b) Imatge contaminada amb un 30% de soroll impulsiu de valor aleatori, i sortides de (c) VMF, (d) NAVMF, (e) IPGF, i (f) IFPGF.

mentre que el soroll ha estat reduït significativament, exceptuant zones on el soroll és molt similar al fons de la imatge.

6.4 Complexitat Computacional

En aquesta secció compararem la càrrega computacional del filtre IFPGF davant del PGF i el VMF. Ja que la operació computacional més demandada és el càlcul de distàncies entre vectors de color, compararem el nombre mitjà de distàncies calculades necessàries en aquest algorisme per a processar una imatge de grandària $N \times N$, encara que aquesta grandària no és rellevant per a les conclusions.

Siga η el nombre de veïns del píxel central en una finestra de filtratge W (per a finestres $n \times n$, $\eta = n^2 - 1$). Primer, determinarem el nombre mitjà de distàncies necessàries per a diagnosticar el píxel central en W . Per això, anomenem r la probabilitat que un píxel \mathbf{x}_j del veïnat satisfaci $\|\mathbf{x}_j - \mathbf{x}_i\| \leq d$ quan el píxel central \mathbf{x}_i és *no corrupte*. Noteu que r serà $r < 1 - p$. Aquestes probabilitats depenen de l'estructura de la imatge, les estadístiques de soroll i intensitat i també del paràmetre distància d .

Una cota superior del nombre de distàncies en la primera fase del procés de detecció en l'algorisme que presentem ve donada per

$$\eta \left(\frac{N}{n}\right)^2.$$

Ara, el nombre de píxels diagnosticats com a no corruptes en aquesta primera fase és

$$(1 - p) \left(\frac{N}{n}\right)^2 (\eta r + 1)$$

i el nombre de píxels diagnosticats com provisionalment corruptes és

$$p \left(\frac{N}{n}\right)^2.$$

Així, el nombre de píxels no corruptes que s'han de diagnosticar a la segona fase és

$$(1 - p)N^2 - (1 - p) \left(\frac{N}{n}\right)^2 (\eta r + 1) = (1 - p)N^2 \left(1 - \frac{1}{n^2}(\eta r + 1)\right)$$

i el nombre de píxels provisionalment corruptes que han de diagnosticar-se a la segona fase és

$$pN^2 - p \left(\frac{N}{n}\right)^2 = pN^2 \left(1 - \frac{1}{n^2}\right).$$

Tret dels resultats de les nostres experiències (consulteu Taula 6.11), a la segona fase podem assumir que els píxels no corruptes que no s'han diagnosticat a la primera fase es classifiquen com no corruptes després d'almenys $2m'$ distàncies.

Taula 6.11. Nombre mitjà de distàncies necessàries per píxel per a un píxel no corrupte a la segona fase del procés de diagnòstic.

	Imatges			
p	Caps	Flower	Girl	Statue
0.10	1.63	1.86	1.46	1.72
0.20	1.64	1.87	1.53	1.73
0.30	1.69	1.89	1.68	1.80
0.40	1.79	1.97	1.82	1.92

Llavors, una cota superior del nombre de distàncies calculades per a diagnosticar tots els píxels a la segona fase ve donada per

$$(1-p)N^2 \left(1 - \frac{1}{n^2}(\eta r + 1)\right) 2m' + pN^2 \left(1 - \frac{1}{n^2}\right) \eta$$

Ara, el nombre mitjà de distàncies calculades a l'algorisme complet per a diagnosticar un píxel de la imatge és

$$c_d^{**} = \frac{\eta \left(\frac{N}{n}\right)^2 + (1-p)N^2 \left(1 - \frac{1}{n^2}(\eta r + 1)\right) 2m' + pN^2 \left(1 - \frac{1}{n^2}\right) \eta}{N^2} =$$

$$= \frac{\eta}{n^2} + (1-p) \left(1 - \frac{1}{n^2}(\eta r + 1)\right) 2m' + p \left(1 - \frac{1}{n^2}\right) \eta$$

Finalment, podem comparar el nombre mitjà de distàncies necessàries per al filtratge d'un píxel en l'algorisme proposat i els descrits a [105] i [12]. La Taula 6.12 mostra les comparatives per als casos $p = 0.1$, $r = 0.8$, $q = 0.01$.

Taula 6.12. Resultats comparatius del nombre mitjà de distàncies necessàries per al filtratge d'un píxel. Els valors c_d , c_d^* i c_d^{**} corresponen als filtres PGF, FPGF i IFPGF, respectivament.

m	m'	c_d	c_d^*	c_d^{**}
3	1	7.004	4.506	1.920

Com s'observa a la taula anterior i deduït dels nostres estudis experimentals, el guany en termes de la reducció del càlcul de distàncies és d'un 72.5% respecte al FPG i d'un 57.4% respecte del FPGF. Aquestes dades es corresponen amb les obtingudes a les nostres experimentacions, on s'obté una reducció del temps computacional al voltant del 80 al 85%, respecte del filtre FPG i al voltant del 40 al 50% respecte del filtre FPGF, depenent de la intensitat de soroll.

6.5 Implementació *fuzzy* del Mètode Proposat

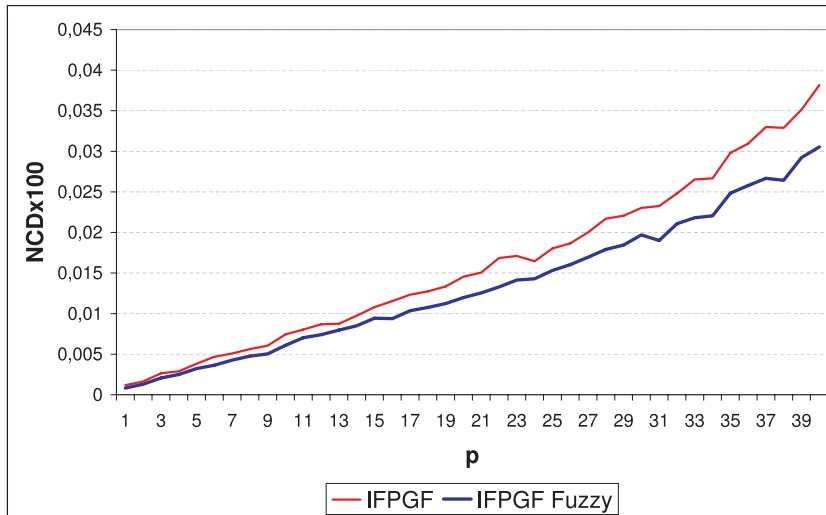
Al llarg del capítol hem desenvolupat un algorisme que obté uns resultats objectius de qualitat molt positius. Com s'ha comentat el treball s'ha centrat al mètode de detecció de píxels sorollosos, obtenint tot i això uns resultats competitius de velocitat. Com que al llarg de la Tesi Doctoral l'objectiu fonamental és la millora computacional dels filtres desenvolupats, no podem deixar passar l'opció de efectuar una implementació amb la mètrica *fuzzy* i estudiar els resultats.

La mètrica *fuzzy* emprada és la ja estudiada al llarg de la Tesi, que recordem a continuació:

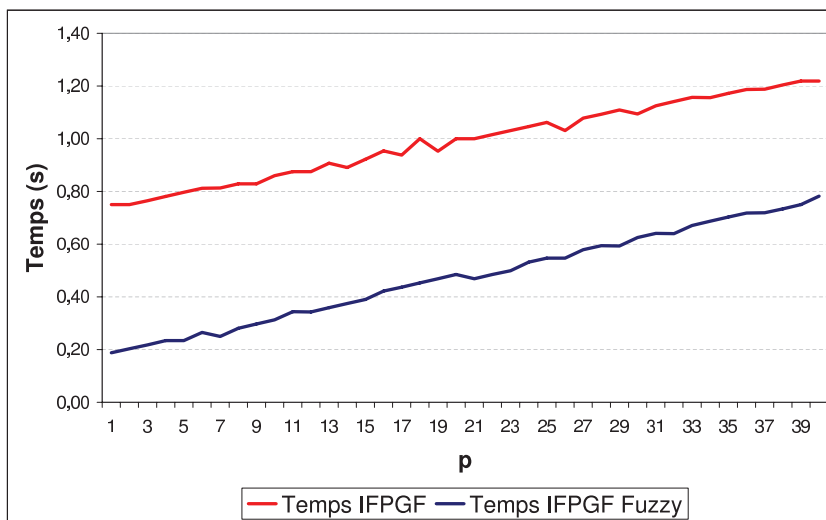
$$M(\mathbf{x}_i, \mathbf{x}_j) = \prod_{l=1}^3 \frac{\min\{x_i(l), x_j(l)\} + K}{\max\{x_i(l), x_j(l)\} + K} \quad (6.1)$$

on x_i i x_j són dos vectors de colors entre els píxels i i j .

A continuació es presenten les figures 6.14 a 6.17 on es veu el rendiment en termes de qualitat del filtre IFPGF i de rendiment computacional amb les seues implementacions amb mètriques L_2 i *fuzzy*. La comparativa en termes de qualitat s'ha centrat exclusivament en el NCD, ja que és la mesura objectiva que indica la qualitat visual de la imatge filtrada.

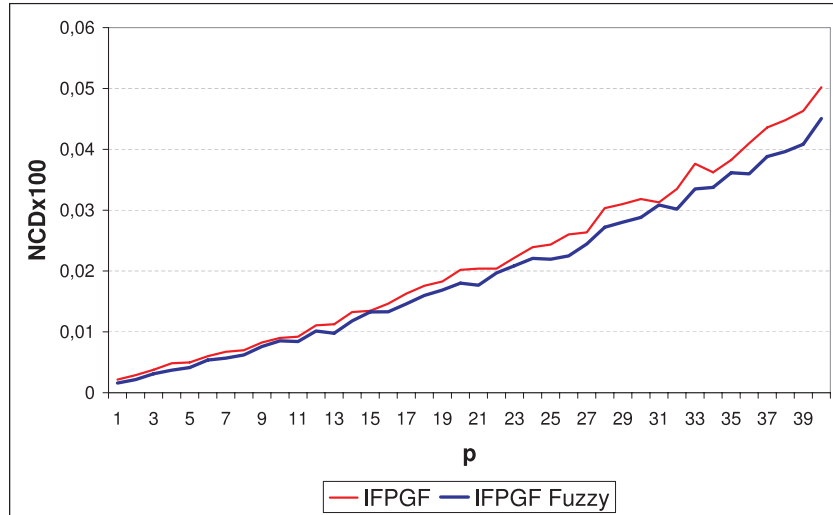


(a)

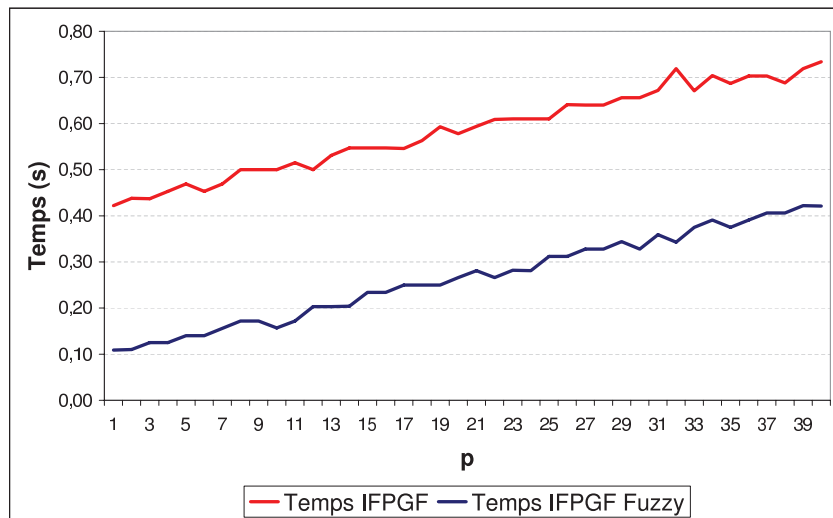


(b)

Figura 6.14. Comparatives per a la imatge Caps: (a) NCD, (b) Temps.

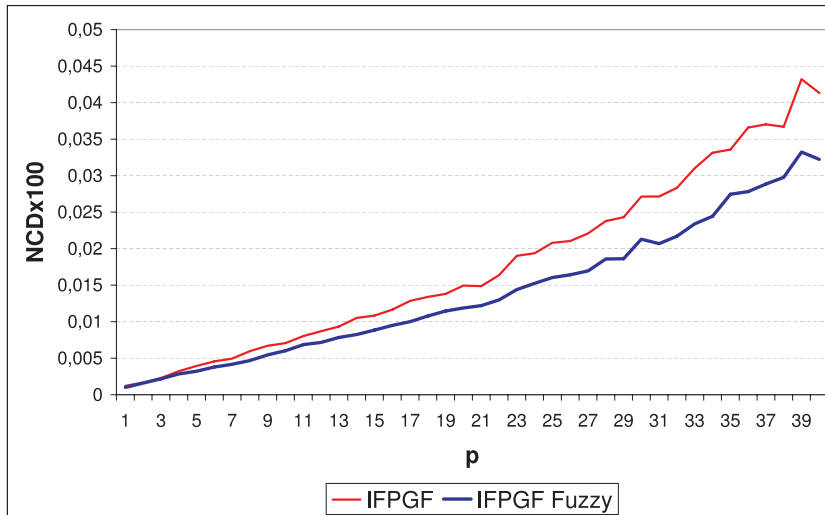


(a)

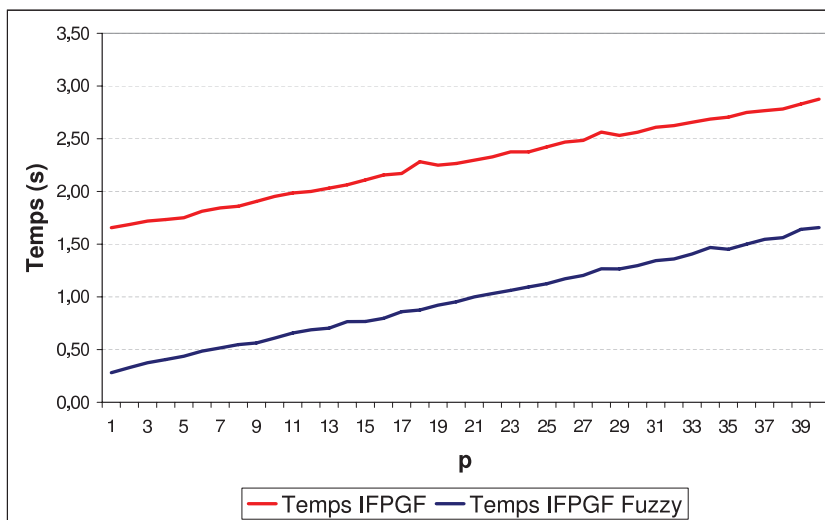


(b)

Figura 6.15. Comparatives per a la imatge Flower: (a) NCD, (b) Temps.

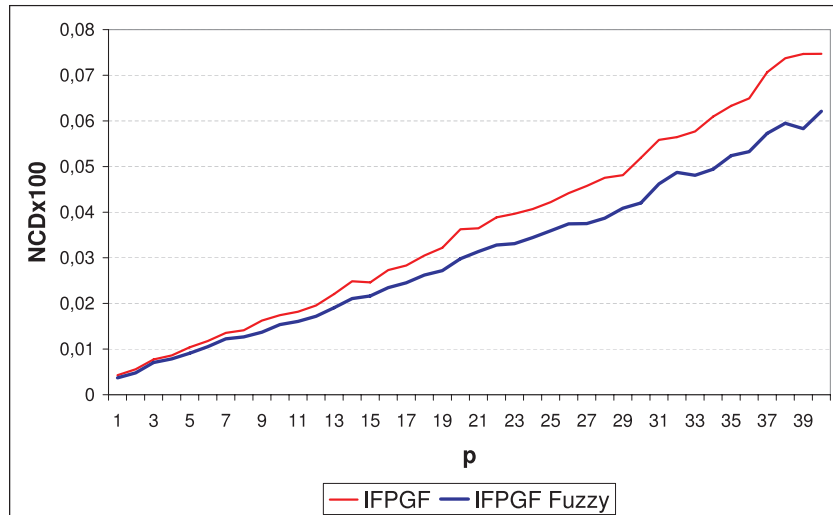


(a)

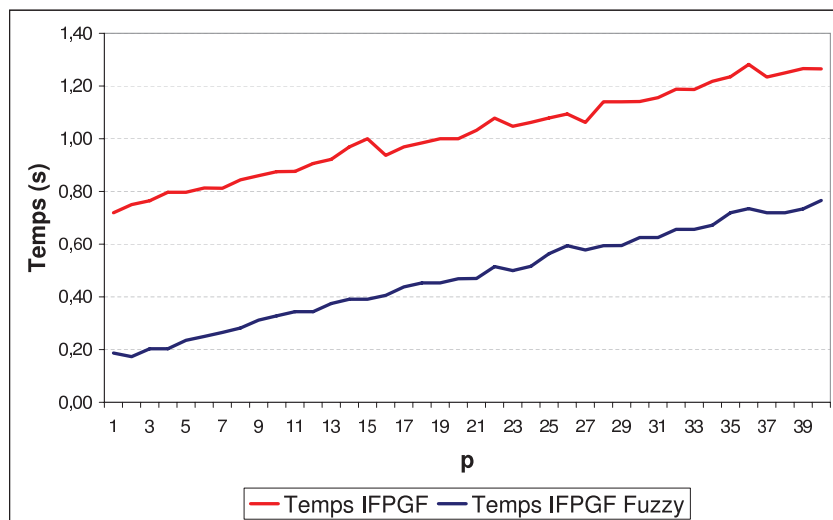


(b)

Figura 6.16. Comparatives per a la imatge Girl: (a) NCD, (b) Temps.



(a)



(b)

Figura 6.17. Comparatives per a la imatge Statue: (a) NCD, (b) Temps.

Per tal de entrar una mica més en detall en les dades, ací presentem dos taules de dades amb les comparatives de temps i NCD per a les imatges Caps i Girl (Taulas 6.13 i 6.14, respectivament).

p	$NCD \times 10^{-2}$		Temps (s)	
	IFPGF	IFPGF Fuzzy	IFPGF	IFPGF Fuzzy
5	0.382	0.321	0.953	0.265
10	0.744	0.608	1.000	0.328
15	1.079	0.944	0.954	0.406
20	1.454	1.196	1.032	0.516
25	1.802	1.531	1.094	0.578
30	2.301	1.969	1.157	0.671
35	2.981	2.484	1.344	0.751
40	3.816	3.055	1.312	0.828

Taula 6.13. Comparativa de qualitat i de rendiment a la imatge Caps dels filtres IFPGF vs IFPGF amb mètrica fuzzy.

p	$NCD \times 10^{-2}$		Temps (s)	
	IFPGF	IFPGF Fuzzy	IFPGF	IFPGF Fuzzy
5	0.393	0.321	2.142	0.594
10	0.707	0.602	2.312	0.781
15	1.083	0.884	2.484	0.984
20	1.494	1.119	2.641	1.187
25	2.079	1.605	2.828	1.406
30	2.713	2.129	2.985	1.609
35	3.358	2.745	3.187	1.813
40	4.131	3.223	3.406	2.031

Taula 6.14. Comparativa de qualitat i de rendiment a la imatge Girl dels filtres IFPGF vs IFPGF amb mètrica fuzzy.

A la vista del resultats torna a quedar de manifest que les mètriques fuzzy són més que adequades per al procés de filtratge d'imatges digitals en color, no solament pel seu cost computacional sinó per la millora en la qualitat de les imatges filtrades.

La comparativa visual de les imatges de sortida es presenta a les Figures 6.18 a 6.22, on s'observa el guany en quant a qualitat de les imatges de sortida respecte a la implementació amb la mètrica L_2 . Pot observar-se que el soroll que no s'elimina en la implementació fuzzy tampoc ha estat eliminat amb la vessant L_2 , mentre que hi ha molt de soroll que no s'elimina amb la mètrica

L_2 que si que s'ha eliminat de manera eficient amb la mètrica fuzzy. Fixar-se per exemple en la zona homogènia central de Caps, la zona de la persiana de Flower així com la cara de Statue.

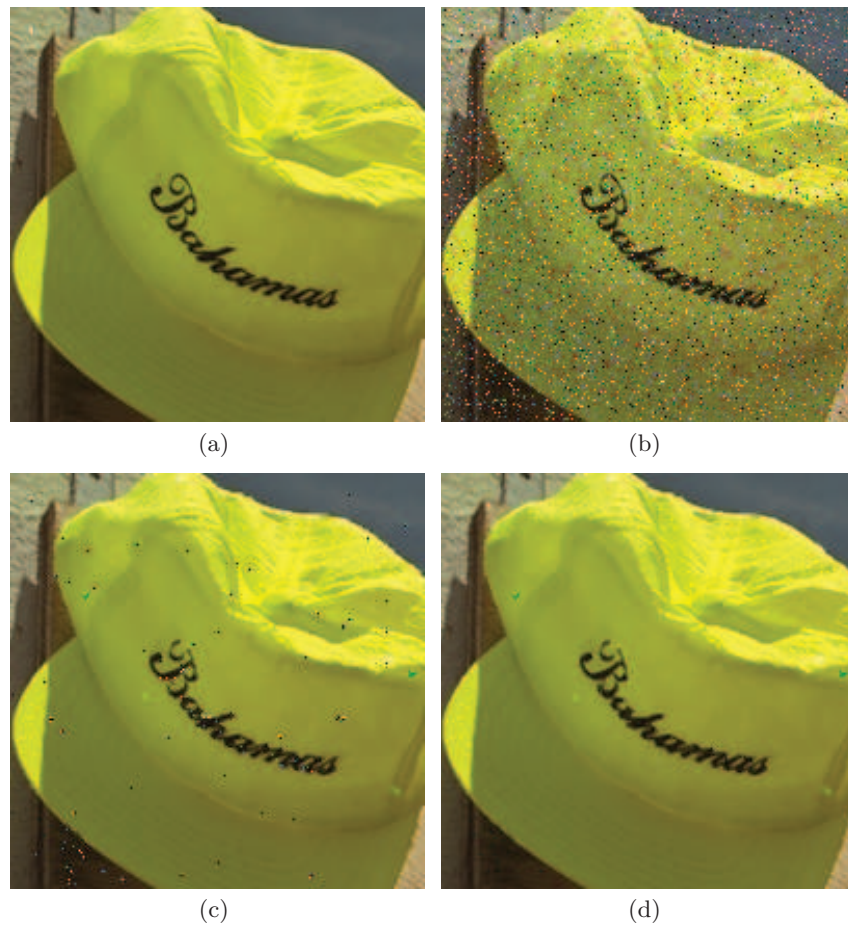


Figura 6.18. Comparatives per a la imatge Caps: (a) Imatge Caps original, (b) imatge Caps contaminada amb un 20% de soroll impulsiu. (c) sortida del filtratge IFPGF, (d) sortida de la implementació del IFPGF *fuzzy*.

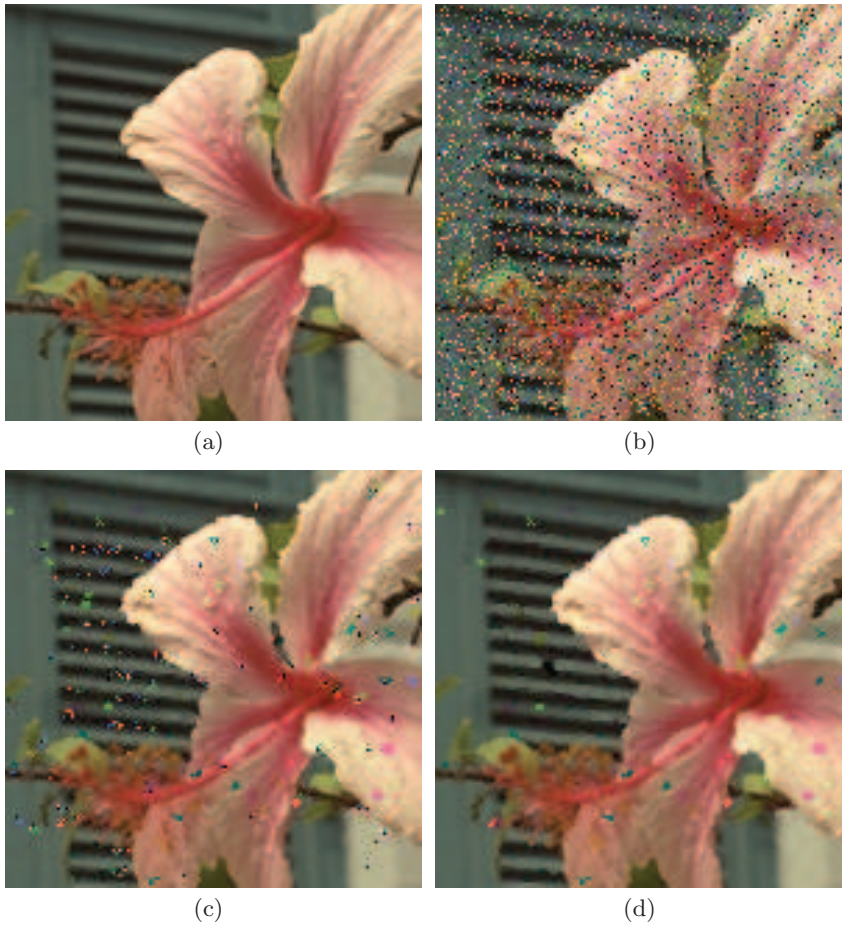


Figura 6.19. Comparatives per a la imatge Flower: (a) Imatge Flower original, (b) imatge Flower contaminada amb un 40% de soroll impulsiu. (c) sortida del filtratge IFPGF, (d) sortida de la implementació del IFPGF *fuzzy*.

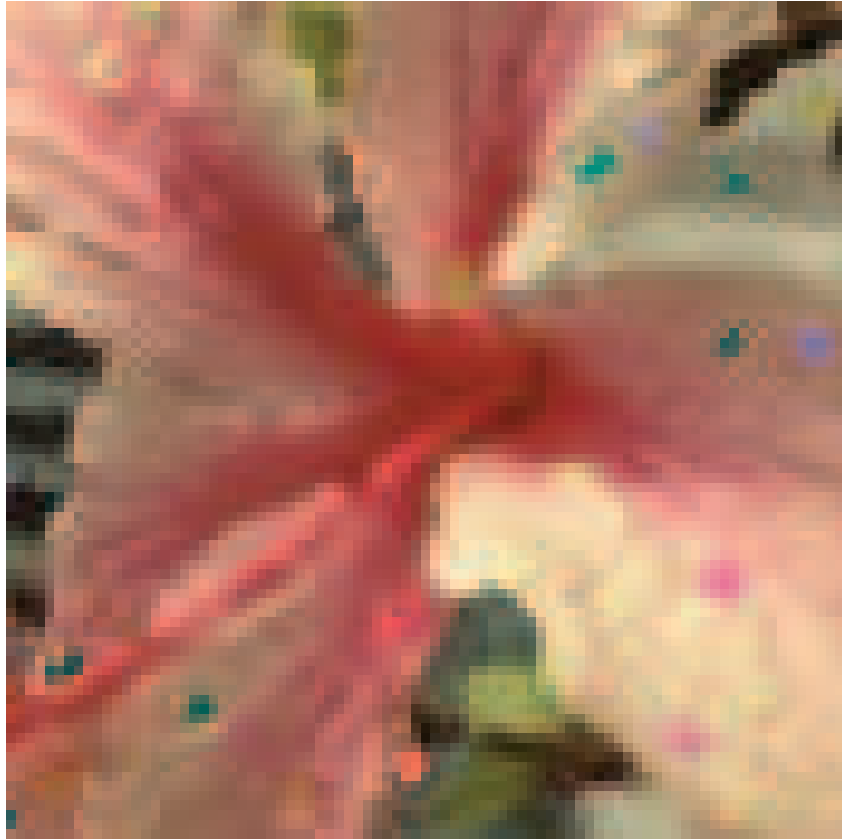


Figura 6.20. Ampliació de la imatge de sortida Flower amb la vessant fuzzy. En aquesta imatge s'observa que molts dels píxels no detectats com a soroll és a causa de cúmuls de píxels sorollosos amb els mateixos valors, amb la qual cosa l'algorisme detecta que en tenen de propers amb les mateixes característiques i per això els marca com *no corruptes* i no els filtra.



Figura 6.21. Comparatives per a la imatge Girl: (a) Imatge Girl original, (b) imatge Girl contaminada amb un 10% de soroll impulsiu. (c) sortida del filtratge IFPGF, (d) sortida de la implementació del IFPGF *fuzzy*.

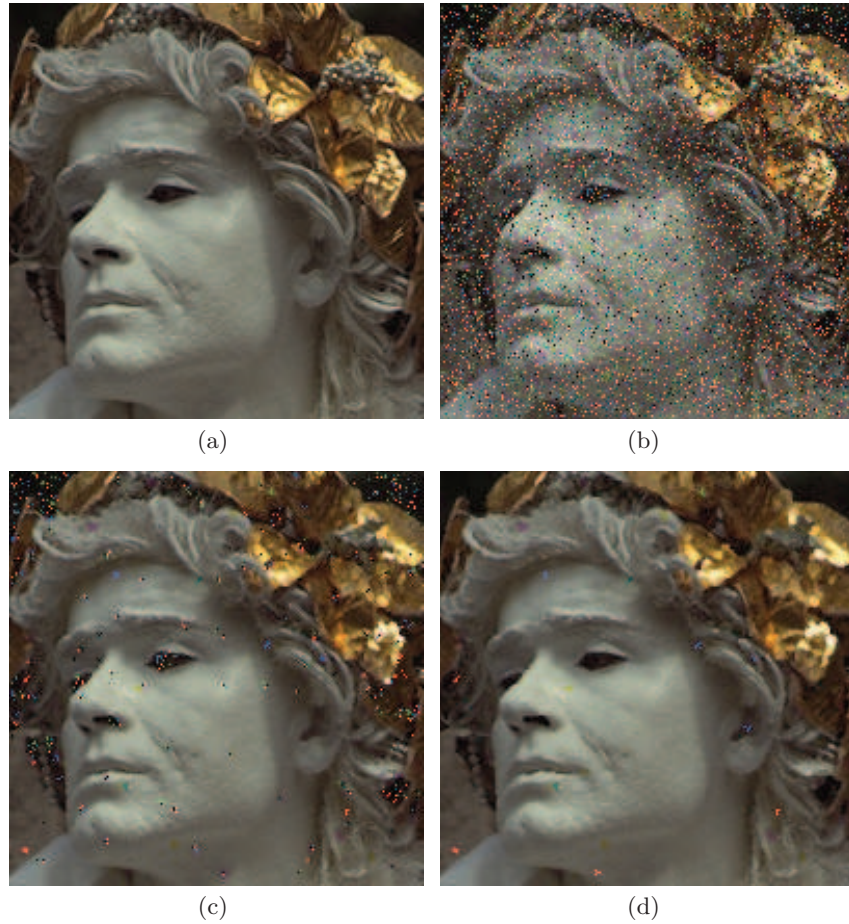


Figura 6.22. Comparatives per a la imatge Statue: (a) Imatge Statue original, (b) imatge Statue contaminada amb un 30% de soroll impulsiu. (c) sortida del filtratge IFPGF, (d) sortida de la implementació del IFPGF *fuzzy*.

6.6 Conclusions

En aquest capítol hem introduït una nova tècnica per a la detecció de soroll impulsiu en imatges en color. La tècnica fa servir el concepte de *peer group* amb la mètrica L_2 i constitueix un desenvolupament avançat de filtres anteriors basats en *peer group*. El mètode proposat fa ús d'una tècnica que accelera la velocitat de detecció juntament a una depurada tècnica en la detecció de soroll. Els estudis experimentals han mostrat que el mètode exhibeix un rendiment que és significativament millor respecte a filtres vectorials recents. S'ha efectuat l'anàlisi del mètode per a comprovar la seua eficiència com-

putacional. Posteriorment s'ha experimentat la mateixa tècnica de filtratge substituint la mètrica L_2 per la mètrica fuzzy M desenvolupada al llarg de la Tesi Doctoral, obtenint resultats que no fan més que confirmar el bon funcionament computacional i l'adequació de les mètriques fuzzy per a la detecció de soroll impulsiu.

En conseqüència, podem concloure que el mètode proposat és robust, precís i eficient i sembla adequat per a aplicacions reals, a causa del seu bon rendiment computacional.

7 Filtratge d'Imatges Mitjançant Computació Paral·lela amb MPI

Resum

Es presenta al capítol actual una implementació paral·lela de les tècniques de filtratge clàssiques i recents d'imatges en color. La computació paral·lela es fa servir per a millorar els temps de resposta de filtratges per a imatges d'alta resolució com per exemple grans angulars o imatges de satèl·lit. S'implementen els filtres VMF, PGF, FMNF i IFPGF per veure el seu comportament i s'estudia en quines línies de treball pot seguir investigant-se en tal de millorar les velocitats de procés en aquests tipus d'imatges.

7.1 Introducció

En els anteriors capítols hem estat desenvolupant nous algorismes i tècniques per al filtratge d'imatges en color amb l'objectiu d'aconseguir tant guanys computacionals (millores de velocitat de filtratge) com millores en la qualitat del filtratge (aconseguir imatges filtrades el més similar possible a les originals).

Els filtres desenvolupats com s'ha vist al llarg de la Tesi Doctoral han donat uns resultats fructífers en les dos línies d'investigació treballades. La qüestió emergent va ser la següent; tenim unes tècniques de filtratge que donen molt bons resultats, llavors, què passaria si s'aplicaren a imatges de grans dimensions o amb unes intensitats de soroll que requerisquen molt de còmput? La resposta va ser que aconseguíem temps de resposta/qualitats millors que en altres filtres similars, però si la imatge a processar és molt gran apareix un requeriment de càlcul independent del filtre, ja que depèn únicament de l'estructura de la imatge.

En aquest moment ens plantejàrem l'ús de la computació paral·lela mitjançant el pas de missatges entre processors amb MPI, i fer un estudi de com milloren el rendiment les tècniques paral·leles. La idea principal és la de "*dividir i vèncer*". L'esquema de dividir i vèncer és una tècnica per resoldre problemes que consisteix a dividir el problema original en subproblemes (de menor grandària o *talla*¹), resoldre els subproblemes i combinar les solucions del subproblemes per a donar una solució del problema original. Si els subproblemes són similars a l'original, llavors poden fer-se servir tècniques de

¹ Es denota per *talla* d'un problema el seu cost computacional

recursivitat. Cada nivell del procediment del paradigma de dividir i vèncer es descomposa en tres passos:

1. Dividir el problema en subproblemes.
2. Resoldre els subproblemes.
3. Combinar les solucions dels subproblemes per obtenir la solució del problema original.

Per què aquestes tècniques funcionen, han de complir-se els dos requisits següents:

1. Els subproblemes han de ser de talla més menuda que l'original.
2. Hi ha un nombre finit de subdivisions per trobar subproblemes resolubles de forma directa.

Les tècniques de filtratge d'imatges implementades anteriorment compleixen els requisits del paradigma de dividir i vèncer, amb la qual cosa partint d'una imatge molt gran es divideix en diversos trossos que es reparteixen entre els processos. Aquests processos filtren la seua part corresponent i retornen la imatge filtrada que serà recomposada formant la solució final.

Evidentment, és lògic que una implementació paral·lela és millor, ja que per a efectuar qualsevol tasca, millor distribuir la feina que fer-la tot sol. Llavors, per què fer aquestes implementacions? La resposta la podem dividir en dos:

- La implementació de filtratge d'imatges amb tècniques paral·leles és una tècnica novedosa.
- Estudiar el comportament no solament del les tècniques de filtratge sinó de les tècniques de distribució de dades per efectuar el paral·lisme

7.1.1 La Necessitat de Potència Computacional

La recerca de més potència ha sigut un dels fils conductors en el desenvolupament d'ordinadors. Quan els científics encara no acabaven d'adaptar-se al centenars d'operacions per segon, ja en demandaven milers, i quan en tenien milers ja en volien milions (i actualment bilions 10^9 i trillions 10^{12})². La pregunta és, per què eixa demanda de potència computacional?

Durant segles, la ciència s'ha basat en el paradigma de primer observar i després teoritzar per a llavors comprovar la teoria mitjançant l'experimentació. En enginyeria el procés és dissenyar i construir prototips i fer els tests. D'altra banda, està fent-se menys costós construir simulacions per computador detallades que fer una sèrie d'experiments o desenvolupar prototips. És a dir, està reemplaçant-se l'experimentació i observació del paradigma científic i el disseny i prototipat del paradigma enginyer per la computació. Més encara, hi ha fenòmens que sols poden modelar-se mitjançant l'ús computacional, com per exemple l'evolució de l'Univers.

² Seguint la notació Americana. En notació Europea 1 bilió= 10^{12}

Vegeu un exemple on suposem que volem calcular la predicció del temps sobre tota Europa ($\approx 10 \times 10^6 km^2$) en els 20 primers quilòmetres de l'atmosfera cada hora dels propers dos dies. Per això farem el càlcul en una malla on cada cub té $0.1 km$. Llavors tenim un total de:

$$10 \times 10^6 km^2 \cdot 20 km \cdot 10^3 \text{ punts per } km^3 = 2 \times 10^{11} \text{ punts}$$

Si suposem que per al càlcul de la predicció són necessaris 100 càlculs durant 48 hores tenim un total de:

$$2 \times 10^{11} \text{ punts} \cdot 10^2 \text{ càlculs} \cdot 48 = 96 \times 10^{13} \simeq 1 \times 10^{15} \text{ càlculs}$$

Si tenim en compte un computador capaç de fer un bilió de càlculs per segon tardarem:

$$1 \times 10^{15} \text{ càlculs} / 10^9 \text{ càlculs per segon} = 10^6 \text{ segons} \simeq 12 \text{ dies}$$

Llavors per calcular la previsió dels propers dos dies en necessitem dotze!! I això que solament ens hem centrat en el cost temporal, sense tenir en compte el cost espacial (consum de memòria) d'aquest tipus de problemes, ja que encara que si l'ordenador és molt ràpid i no disposa de la suficient capacitat d'emmagatzematge, el coll de botella que li provoca eixa carència de memòria li fa perdre aquesta potència. La conclusió és que necessitem més potència (velocitat i emmagatzematge) per afrontar problemes d'aquest estil, com per exemple algorismes de xifrat de cadenes d'ADN, simulacions de bio-mol·lècules, etc.

7.1.2 La Necessitat de la Computació Paral·lela

Suposem que disposem d'un computador que pot executar el següent bloc de codi en un segon:

```
// x, y, z són arrays de reals, cadascun dels quals conté un trilió d'elements
for ( i=0; i<UN_TRILIO; i++)
    z [ i]=x [ i]+y [ i ];
```

això ens implica portar $x[i]$ i $y[i]$ de memòria a la CPU i guardar-ho a $z[i]$, el que són 3×10^{12} còpies entre memòria i CPU. Seguint l'estudi vist a [79], això seria possible sols si una paraula de memòria (32 bits usualment) ocupara físicament 10^{-10} metres, és a dir una mica menys d'un àtom, amb la qual cosa podem dir que encara que trobarem un processador tan ràpid, hi ha sèries limitacions físiques.

Per tot això, el que podem aplicar és el que s'aplicava ja en temps de Juli Cèsar "*divide et vinces*", és a dir repartir la feina entre diversos computadores. Com? Mitjançant un *computador paral·lel*, que és un computador (o conjunt d'aquests) amb múltiples processadors que poden treballar conjuntament en la solució d'un problema.

Amb el fet de tindre el computador no és suficient, ja que un cop tenim la col·lecció de processadors i de memòries, hem de:

1. Decidir una implementació i xarxa d'interconnexió entre els processadors i els mòduls de memòria.
2. Dissenyar i implementar un software que gestione el hardware disponible.
3. Cercar algorismes i estructures de dades que solucionen el nostre problema.
4. Dividir l'algorisme i estructures de dades en subproblemes.
5. Identificar el mode de comunicació entre els subproblemes.
6. Assignar subproblemes a processadors.

En aquest capítol, per aconseguir la implementació paral·lela hem fet servir MPI (Message Passing Interface), que simplement són unes funcions de llibreria que poden incloure's en programes en C o en Fortran. El pas de missatges és el mètode més emprat per a construir programes paral·lels, amb una eficiència més que raonable.

7.1.3 Introducció al Filtratge d'Imatges en Color

El filtratge d'imatges digitals en color és una de les tasques més demandades a causa de l'ampli catàleg d'aplicacions on estan presents. Els filtres vectorials clàssics com les tècniques recents treuen profit de l'estadística robusta. Un dels problemes de les tècniques de filtratge és el cost computacional que requereixen per a efectuar el filtratge. Aquest cost si l'apliquem a imatges menudes no és gaire apreciable, però si ens centrem ja a l'àmbit de fotografia d'alta resolució, imatges de satèl·lit i similars, aquest cost s'incrementa desmesuradament.

Altre punt d'interès és el comportament de les tècniques de filtratge en quant a rendiment del filtre. Certes tècniques de filtratge tendeixen a difuminar detalls de les imatges com puguen ser vores, detalls fins o textures, per que filtren o processen tots els píxels de la imatge, tant si hi ha soroll present com si no. Filtres clàssics com VMF [4], DDF [47] i BVDF [114] tenen aquest inconvenient. Els esquemes de filtratge per substitució corregeixen aquest problema, com hem comentat en els capítols anteriors, ja que en ells es realitza una primera tasca de detecció de soroll per a posteriorment filtrar aquells píxels marcats com a sorollosos, aplicant l'operació identitat sobre els píxels lliures de soroll. Entre aquestes tècniques destaquen els esquemes amb *peer group* com per exemple PGF [105], i els presentats recentment FMNF [13] i IFPGF [14].

Es presenta al capítol actual una proposta de filtratge mitjançant computació paral·lela. Per això s'han fet servir les llibreries MPI (*Message Pass Interface*) per a implementar versions d'algorismes recents i clàssics mitjançant MPI. El mètode de filtratge és el mateix que en els filtres mencionats anteriorment, el que s'ha fet és un algorisme de distribució d'informació que és on està el coll de botella de la computació paral·lela. El desenvolupament

dels algorismes, així com els resultats de l'experimentació es presenten a les respectives seccions

El capítol s'organitza com segueix. A la Secció 7.2 es presenta un estudi que mostra com s'optimitza l'emmagatzemament d'imatges en color en memòria. La Secció 7.3 mostra un recorregut per l'estat de l'art actual del filtres vectorials, tant clàssics com recents. Posteriorment, a la Secció 7.4 es mostra l'algorisme de filtratge paral·lel, centrant-se en la distribució de les dades. Es conclou amb les Seccions 7.5 i 7.6 que presenten els resultats de l'experimentació així com les línies a seguir en treballs futurs.

7.2 Representació en Memòria d'Imatges

Una imatge a color RGB es pot considerar com una matriu tridimensional $n \times m \times 3$, sent n l'alt i m l'ample. Cada punt (i, j) , $i = 0, 1, \dots, n - 1$, $j = 0, 1, \dots, m - 1$ representa un **píxel** de la imatge. Per a cada píxel necessitem el seu vector de color RGB que pren valors enters entre $[0, 255]$, corresponent a les intensitats de color. A banda d'aquesta matriu, la imatge incorpora les capçaleres típiques amb metainformació de l'arxiu on es guarda físicament la imatge, informació que no farem servir per al procés de filtratge, sinó per al procés de càrrega en memòria de la imatge.

Un dels problemes que apareix és que com a priori no sabem la grandària de la imatge, per la qual cosa haurem de *reservar dinàmicament memòria* per als colors (les capçaleres són de grandària constant). Així doncs, una primera aproximació per a representar imatges en memòria fent reserva de memòria seria la que es veu al Llistat 7.1. Aquesta versió és molt intuïtiva, però suposa un sobrecost de memòria molt elevat. Aquest sobrecost són els punters que es creen per a adreçar files, columnes i els elements de color de la imatge. Vegeu-ho amb un simple càlcul:

Suposem una imatge de 1.000×1.000 . Així indica que al primer *malloc*³ es reserven 1.000 punters de files per a continuació al primer bucle reservar per a cada punter de fila 1.000 punters de columnes, amb la qual cosa ja tenim 1.000 punters de fila més 1.000.000 de punters de píxels, un total de 1.001.000 punters. Al bucle més intern és reserva memòria per als 3 components RGB de color de cada píxel (3 bytes). Llavors tenim 3.000.000 bytes que ocupen els colors més 4.004.000 que ocupen els punters el que fa un total de 7.004.000 bytes. Noteu que hi ha més consum de memòria fet pels punters que per les dades pròpiament dites.

De manera genèrica (en un llenguatge com C on els punters ocupen 4 bytes), una imatge de $N_1 \times N_2$ píxels per a ser representada com una matriu dinàmica li fan falta N_1 punters fila, per a després reservar $N_1 \times N_2$ punters columna. Això implica un total de $N_1 + N_1 \times N_2$ punters, amb un consum

³ Funció de llibreria que reserva memòria de forma dinàmica.

Llistat 7.1. Versió amb matriu dinàmica

```

typedef unsigned char byte;

typedef enum {R,G,B} Color;

typedef struct {
    char bmTipo[2];
    unsigned int bmTamanyo;
    short bmReserva1;
    short bmReserva2;
    int bmOffset;
} BmpFicheroCabecera;

typedef struct {
    int tamCabecera;
    int ancho;
    int alto;
    short planos;
    short bits;
    int compresion;
    int tamImagen;
    int ppm_h;
    int ppm_v;
    int colores;
    int colores_imp;
} BmpImagenCabecera;

typedef struct{
    BmpFicheroCabecera cabFich;
    BmpImagenCabecera cabImg;
    byte ***imagen;
} BMPImagen;

//Reserva de memoria a l'obrir i un cop llegit el struct
BMPImagen I;
.
.
.
I.imagen=(byte ***)
    malloc (I->cabImg.alto*sizeof(byte **));
for (i=0;i<I.cabImg.alto;i++){
    I.imagen[i]=(byte **)
        malloc (I.cabImg.ancho*sizeof(byte *));
    for (j=0;j<I.cabImg.ancho;j++)
        I.imagen[i][j]=(byte *)
            malloc (3*sizeof(byte));
}

```

Llistat 7.2. Versió amb matriu lineal (vector) dinàmic

```

typedef struct{
    BmpFicheroCabecera cabFich;
    BmpImagenCabecera cabImg;
    byte *imagen;
} BMPImagen;

//Reserva de memoria a l'obrir i un cop llegit el struct
BMPImagen I;
.
.
I.imagen=(byte *)
    malloc(I.cabImg.alto*I.cabImg.ancho*3*sizeof(byte **));

```

Llistat 7.3. Accés als elements de la imatge com una matriu

```

// accés a les dades de la matriu
// imprimim la imatge
for (i=0;i<I.cabImg.alto-1;i++){
    for (j=0;j<I.cabImg.ancho-1;j++){
        printf("\d",I.imagen[i][j][R]);
        printf("\d",I.imagen[i][j][G]);
        printf("\d",I.imagen[i][j][B]);
    }
    printf("\n");
}

```

de $(N_1 + N_1 \times N_2) \times 4$ bytes. Posteriorment per a la imatge en sí (els píxels) són necessaris $N_1 \times N_2 \times 3$ bytes més, amb un total de $4N_1 + 7N_1N_2$ bytes.

La mateixa imatge $N_1 \times N_2$ implementada com un *array* necessita un sol punter (4 bytes) i la memòria per als píxels $N_1 \times N_2 \times 3$, necessitant per tant $3N_1N_2 + 4$, amb el consegüent estalvi de recursos espacials.

La solució que s'ha optat en tal de millorar l'eficiència és linealitzar la matriu tridimensional de manera que amb un sol punter farem referència a tota la matriu, com es veu al llistat 7.2. D'aquesta manera la despesa de memòria per a la imatge de l'exemple anterior sols és de 3.000.004 bytes (els 4 corresponen al punter).

Evidentment hi ha un desavantatge, l'accés no és tan senzill com a una matriu estàndard, però amb uns ajusts adequats, l'accés és equivalent, com es veu als llistats 7.3 i 7.4.

Llistat 7.4. Accés als elements de la imatge amb un únic punter

```

// accés a les dades de la matriu com un array
// imprimim la imatge
for ( i=0; i<I.cabImg.alto-1; i++){
    for ( j=0; j<I.cabImg.ancho-1; j++){
        printf( "\d" ,*(I.imagen+(i*ancho+j)*3+R));
        printf( "\d" ,*(I.imagen+(i*ancho+j)*3+G));
        printf( "\d" ,*(I.imagen+(i*ancho+j)*3+B));
    }
    printf( "\n" );
}

```

7.3 Filtratge d'Imatges

A aquesta secció farem una revisió dels algorismes de filtratge d'imatges com a tal sense tenir en compte la paral·lització, que es veurà a la Secció posterior. Dins dels filtres implementats hi ha de dos tipus, els filtres d'una única fase i de dos fases. Explicarem breument el funcionament de cadascun d'ells.

7.3.1 Filtres d'Una Fase

Vector Median Filter - VMF Aquest és dels primers algorismes de filtratge d'imatges [4]. Donada una grandària de finestra corredera senar (habitualment 3), es fa una anàlisi de quin dels píxels de la finestra és el més adequat per a ocupar la posició central. Recordem que el procés és realitzat calculant la suma de les distàncies de cada píxel a la resta dels píxels de la finestra, i l'element amb la mínima suma acumulada és el que passarà a ocupar la posició central de la finestra.

El problema que presenta aquest filtratge és que el procés s'aplica a tots els píxels, tant sorollosos com bons. El resultat és que a banda de suprimir el soroll introdueix a la imatge gran quantitat d'emborronament, suavitzat de vores i altres efectes no desitjats.

Altre problema és la gran quantitat de càlculs que s'han de fer per al filtratge, ja que en certs moments de l'algorisme s'arriben a niuar 6 bucles.

Arithmetic Mean Filter - AMF Aquest filtre [84], és semblant a l'anterior en el mecanisme però distint en quant a l'operació de decidir quin és el nou píxel central. Així com al VMF se substituïa el píxel central per aquell més adequat, al AMF es calcula el píxel central per la mitjana aritmètica dels veïns.

Segueix presentant el mateix problema de l'emborronament de textures i vores que el VMF, a l'igual que es filtren tots els píxels, a banda que el

nou píxel central és calculat (en compte de ser un existent). L'avantatge que presenta és un millora computacional ja que es fan menys càlculs que al VMF.

7.3.2 Filtres de Dos Fases

Com ja hem vist a l'apartat anterior, el gran inconvenient dels algorismes anteriors d'una fase és la introducció de l'emborronament que comentàvem així com el filtratge de tots els píxels, sorollosos o lliures de soroll. Per tal de previndre aquests inconvenients aparegueren els filtratges en dos fases. El seu funcionament és similar en quan a concepte, el que canvia són les distintes implementacions. La idea subjacent en tots els dissenys és el següent:

- 1^a Fase Detecció de soroll mitjançant qualsevol algorisme. Rep com a entrada la imatge sorollosa i dona com a sortida una imatge de les mateixes dimensions amb 1's i 0's anomenada *matriu de soroll*. La existència de un 1 al píxel (i, j) vol dir que és sorollós.
- 2^a Fase Neteja de la imatge sorollosa. Rebut com a entrada la imatge sorollosa i la matriu de soroll es processen solament els píxels que estan a 1 en la matriu de soroll. Així ja aconseguim filtrar únicament els píxels sorollosos. L'algorisme per a netejar-lo pot ser o bé el VMF o AMF.

Peer Group Filter - PGF El *Peer Group Filter* [105], com hem comentat és un filtre de dos fases. Aquest algorisme per a començar, a banda de la imatge a filtrar necessita els següents paràmetres:

1. m és el nombre mínim de veïns que ha de tindre un píxel per a ser declarat lliure de soroll. A partir d'ací es considera que si el *peer group* té almenys $m + 1$ membres (contant el central) pot diagnosticar-se el central com lliure de soroll.
2. d és la distància a la que com a molt s'han de trobar els veïns. La distància es calcula amb la mètrica L_2 (o *Euclídea*).
3. w és la grandària de vora de la finestra d'anàlisi.

Segons se cita a [105], els paràmetres considerats com a robusts són $m = 2$, $d = 45$, $w = 3$ i són els que farem servir a les nostres variants paral·leles.

Vegem el funcionament de cada fase:

- 1^a Fase Sobre cada píxel de la imatge es construeix una finestra d'anàlisi (finestra corredera) de grandària $w \times w$. A continuació es recorren els veïns calculant la distància d'aquests al central. En cas que trobar-ne m veïns la distància dels quals al central siga menor a d es declara el píxel central com lliure de soroll, sent corrupte en cas contrari. Tota aquesta informació ve referida en la que s'anomena una matriu de soroll, inicialment a zeros on un 1 indica que el píxel i, j és sorollós.
- 2^a Fase Tenint com a entrada la imatge a filtrar i la matriu de soroll calculada a la fase 1 es processen els píxels sorollosos. Per a cada píxel sorollós

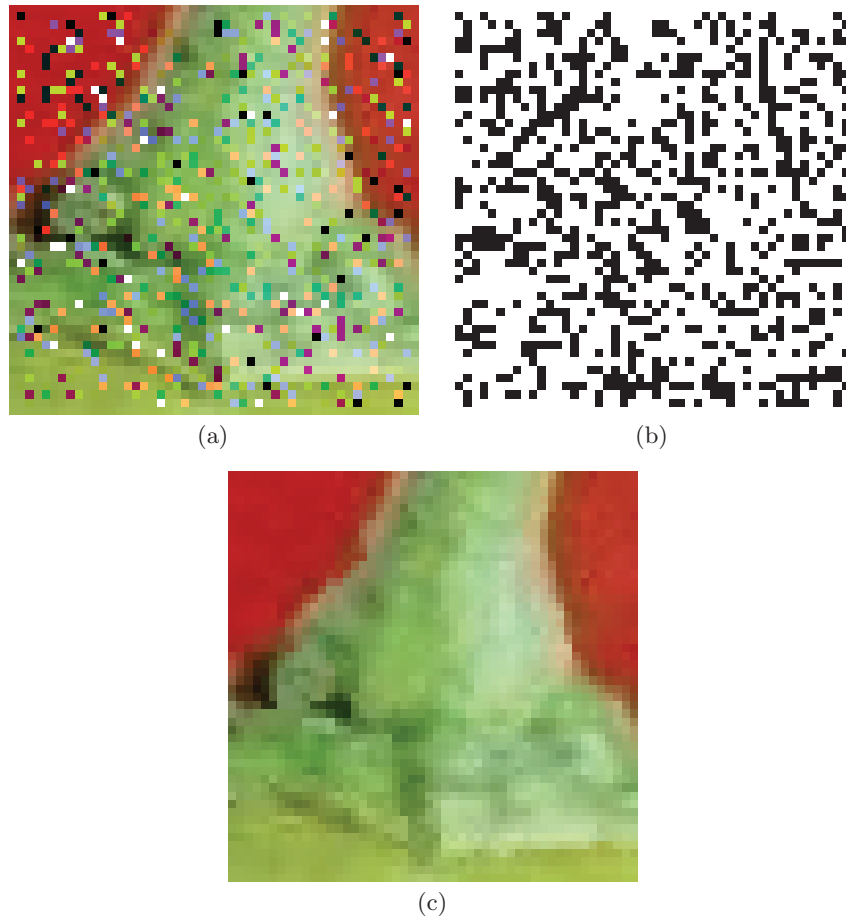


Figura 7.1. Filtratge en dos fases (IFPGF): (a) Imatge de Peppers amb un 30% de soroll, (b) Matriu de soroll (píxels negres), (c) Sortida del filtre (AMF als píxels sorollosos).

es calcula la mitjana aritmètica dels seus veïns no corruptes dins de la finestra $w \times w$. En el cas que dins la finestra de filtratge no hi haja píxels bons aquesta s'incrementaria la grandària de la mateixa fins trobar-ne almenys un.

Fuzzy Modified Neighbourhood Filter - FMNF *Fuzzy Modified Neighbourhood Filter* [13] és un filtre també de dos fases. Els paràmetres necessaris són:

1. m és el nombre mínim de veïns que ha de tindre un píxel per a ser declarat lliure de soroll.

2. d és distància a la que com a molt s'han de trobar els veïns. La distància es calcula amb la mètrica *fuzzy* desenvolupada a [92].
3. w és la grandària de la finestra d'anàlisi.

Segons se cita a [13], els paràmetres robusts són $m = 2$, $d = 0.95$, $w = 3$. Les fases són:

- 1^a Fase Comencem a recórrer cada píxel de la imatge, on es construeix una finestra d'anàlisi (finestra corredera) de grandària $w \times w$. A continuació es recorren els veïns calculant la distància d'aquests al central. En cas de trobar-ne m veïns la distància dels quals al central siga major a d (recordeu que la mètrica és *fuzzy*) es declara el píxel central com lliure de soroll, però també els seus veïns, seguint la desigualtat triangular de la mètrica *fuzzy*. Si la distància és inferior a d es marca el central com corrupte. Quan passem al següent píxel, si aquest ja ha estat declarat com lliure de soroll al pas anterior no fem l'anàlisi i passem al següent. Aquesta informació s'emmagatzema també a la matriu de soroll, inicialment a zeros on un 1 indica que el píxel i, j és sorollós.
- 2^a Fase Tenint com a entrada la imatge a filtrar i la matriu de soroll calculada a la fase 1 es processen els píxels sorollosos. Per a cada píxel sorollós es calcula la mitjana aritmètica dels seus veïns no corruptes dins de la finestra $w \times w$. En el cas que dins la finestra no hi haja píxels bons aquesta s'incrementaria en grandària fins trobar-ne almenys un.

Improved Fast Peer Group Filter - IFPGF [14] Estem davant d'un altre filtre en dos fases. Els paràmetres necessaris són:

1. m és el nombre mínim de veïns que ha de tindre un píxel per a ser declarat lliure de soroll.
2. m' és el nombre mínim de veïns lliures de soroll que ha de tindre un píxel per a ser declarat lliure de soroll.
3. d és la distància a la que com a molt s'han de trobar els veïns. La distància es calcula amb la mètrica L_2 .
4. w és la grandària de la finestra d'anàlisi.

Segons els autors, els valors robusts dels paràmetres són $m = 2$, $m' = 1$, $d = 0.95$, $w = 3$. Les fases són:

- 1^a Fase Es recorre la imatge amb finestres disjunctes de grandària $w \times w$. A continuació es recorren els veïns calculant la distància d'aquests al central. En cas que trobar-ne almenys m veïns la distància dels quals al central siga menor a d , es declara el píxel central com lliure de soroll, així com tots veïns la distància dels quals siga menor a d . Si hi ha menys de m es declara el central com corrupte sense poder afirmar res dels veïns. Noteu que el més probable és que es declaren més de m veïns com a bons per finestra corredera quan el central és bo. A un segon recorregut s'analitzen els píxels que encara no han estat avaluats (veïns llunyans de

píxels centrals bons). La novetat és que per declarar-lo com a bo buscarem o bé m veïns propers o bé m' veïns ja declarats a la primera fase com a bons. Al final tota aquesta informació s'emmagatzema també a la matriu de soroll, inicialment a zeros on un 1 indica que el píxel i, j és sorollós.

2^a Fase Tenint com a entrada la imatge a filtrar i la matriu de soroll calculada a la fase 1 es processen els píxels sorollosos. La mecànica per filtrar cadascun dels píxels sorollosos és calcular la mitjana aritmètica dels seus veïns (sols d'aquells marcats com a bons) dins de la finestra $w \times w$. En cas que dins la finestra no hi haja píxels bons, aquesta s'incrementaria en grandària fins trobar-ne almenys un.

7.4 Paral·lelitzant els Filtres

La gran dificultat que presenta el filtratge paral·lel d'imatges és la divisió d'aquestes per a enviar-les als distints processos. La implementació s'ha realitzat amb MPI, i per tant hem fet servir les seues primitives més bàsiques, principalment `MPI_Send`⁴, `MPI_Recv`⁵ i `MPI_Bcast`⁶. Com s'ha comentat a la secció 7.2 la imatge la tenim emmagatzemada de manera lineal (com un *array*) a memòria. Llavors tenint en compte que un requisit és que el nombre de files de la imatge siga divisible entre p (nombre de processos en que es repartirà la feina), podria pensar-se que amb la primitiva `MPI_Gather`⁷ i `MPI_Scatter`⁸ podríem dividir la imatge per als p processos que hi ha. La resposta és no, ja que si un procés k ha de filtrar n files necessita tenir $n + 2$ files, una per dalt i una per baix tal i com es veu a la figura 7.2.

$i-1, j-1$	$i-1, j$	$i-1, j+1$		
$i, j-1$	i, j	$i, j+1$		
$i+1, j-1$	$i+1, j$	$i+1, j+1$		
		$i-1, j-1$	$i-1, j$	$i-1, j+1$
		$i, j-1$	i, j	$i, j+1$
		$i+1, j-1$	$i+1, j$	$i+1, j+1$

Figura 7.2. Necessitat de files extra

El motiu no és altre que com que s'ha de construir una finestra corredera sobre cada píxel d'anàlisi (excloent vores), les vores són necessàries per a filtrar els píxels al costat de les vores. Això seria ja suficient per al cas de

⁴ Primitiva d'enviament de dades.

⁵ Primitiva de recepció de dades.

⁶ Primitiva d'enviament de dades per difusió o *broadcast*.

⁷ Primitiva de divisió i enviament de dades per a vectors/matrius.

⁸ Primitiva de recepció de dades per a vectors/matrius.

filtratges directes, com per exemple VMF i AMF però no és suficient en els filtratges en dos fases (detecció de soroll + filtratge). La sortida de la primera fase ha de ser l'anomenada *Matriu de Soroll*. La matriu de soroll és una matriu amb les mateixes dimensions que la imatge a filtrar on un 1 vol dir un píxel corrupte, mentre que un 0 vol dir que el píxel és no corrupte. A la segona fase es filtraran els píxels sorollosos fent la mitjana aritmètica dels seus veïns no corruptes, amb la qual cosa necessitarem que la matriu de soroll incloga les vores, i per tant per a la primera fase encara necessitarem dues files extra més, el que fa un total de 4 files extra. Amb un exemple d'una imatge que clarificarà més les coses.

Suposem una imatge de $20 \times Z$ (realment la dimensió de l'amplària, Z , no ens interessa, ja que la imatge la dividirem per files) i un nombre p de processos igual a 4. Amb això, a cada procés li correspon $20/4 = 5$ files a filtrar. Per tant cada procés necessitarà 9 files per a realitzar la tasca del filtratge. Es pot apreciar l'exemple a la figura 7.3.

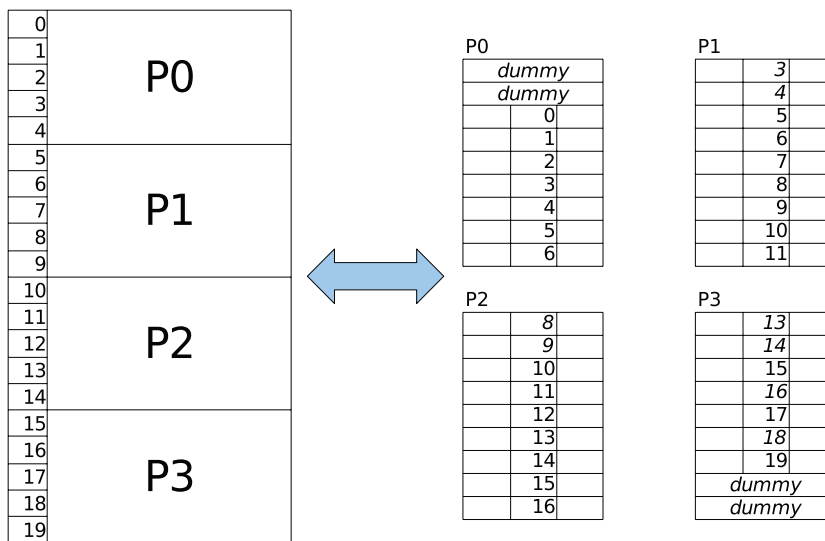


Figura 7.3. Repartiment de files entre els processos

Com que les files 0 i 19 no es filtren per ser vores, les seues files *extra* estan marcades com a *dummy*, ja que no importa el contingut que tinguen (s'ompliran a zeros). Quant als processos "centrals", si prenem com a exemple el procés P1, podem veure que necessita les files 3 i 11 per determinar el soroll de les files 4 i 10 i amb aquestes es podrà filtrar de manera correcta les files 5 i 9 (recordeu que el procés P1 ha de filtrar el rang 5 – 9).

Vistes les necessitats de les files per a cada procés, a l'Algorisme 7 queda descrit el funcionament del filtratge paral·lel.

Algorisme 7: Descomposició i filtratge

Input: ImatgeSorollosa,ParàmetresFiltrat

Data: i,p,Files,Rang,Mr,SubImatge

Output: ImatgeNeta

// Inicialització

```
1 Obtenir p;  
  // p és el nombre de processos  
2 Obtenir Rang;  
  // Rang és el nombre de procés  
3 Si (Rang == 0) Llavors  
  | // Sóc el procés pare  
4  Llegir Imatge; // alt, ample i colors  
5  Si (alt%p! = 0) Llavors  
6  | Sortir;  
7  Broadcast Paràmetres Imatge;  
8  Per A i ← 1 Fins p Fer  
  | // S'envia a cada subprocés la porció d'imatge a  
  |   filtrar  
9  | Reservar memòria SubImatge;  
10 | Copiar dades de Imatge a SubImatge;  
11 | Enviar SubImatge al procés i;  
  |  
  | // El procés 0 efectua el filtratge  
12 | Reservar memòria SubImatge de 0;  
13 | Copiar dades de Imatge a SubImatge de 0;  
14 | Filtrar SubImatge de 0;  
15 | Copiar Subimatge de 0 a Imatge;  
16 | Per A i ← 1 Fins p Fer  
  | | // Es reben les porcions ja filtrades  
17 | | Rebre SubImatge del procés i;  
18 | | Copiar dades de SubImatge a Imatge;  
19 | Guardar Imatge a fitxer;  
20 Si no  
  | // Sóc un procés normal  
21 | Rebre amb broadcast dimensions a processar;  
22 | Reservar memòria per a Subimatge;  
23 | Rebre Subimatge de 0;  
24 | Filtrar Subimatge;  
25 | Enviar Subimatge a 0;  
26 | Alliberar memòria de Subimatge;
```

7.5 Resultats Experimentals

Les experimentacions s'han realitzat al cluster *hyades.upv.es*. Aquest és un cluster IBM 1350 constituït per un conjunt de 60 servidors biprocessador *Intel Xeon* i sistema operatiu *Linux Red Hat*. Està compost de 57 nodes destinats a serveis de càlcul i tres nodes que realitzen tasques fonamentals per al funcionament del cluster.

S'han fet servir per a l'experimentació 3 imatges de grandàries mitjana, gran i molt gran, que es veuen, en aquest ordre, a la figura 7.4.



(a)

(b)



(c)

Figura 7.4. Imatges experimentals: (a) Bee (2192×2700), (b) Kiev (3600×3640), (c) Formentera (4186×9780).

Les experimentacions s'han executat mitjançant les cues del cluster, amb execucions amb 4 processadors i també execucions en mode monoprocessador per poder comparar l'eficiència d l'algorisme. Els resultats poden veure's a les taules 7.1-7.3, i comentem a continuació.

Taula 7.1. Resultats per a la imatge **Bee**

Filtre		Intensitat de soroll impulsiu								
		10,00 %			20,00 %			30,00 %		
		Mono	Multi	%	Mono	Multi	%	Mono	Multi	%
FMNF	Càrrega	1.603	1.583		1.589	1.586		1.622	1.588	
	Distribució	0.000	1.334		0.000	1.336		0.000	1.334	
	Filtratge	2.327	1.836		2.437	1.952		3.002	2.061	
	Total	3.930	4.753	-20.94	4.025	4.874	-21.10	4.624	4.981	-7.72
	P0		0.433			0.549			0.642	
	P1		0.455			0.584			0.826	
	P2		0.475			0.589			0.684	
	P3		0.448			0.558			0.666	
IFPGF	Càrrega	1.588	1.587		1.586	1.585		1.600	1.589	
	Distribució	0.000	1.335		0.000	1.334		0.000	1.335	
	Filtratge	2.931	2.039		4.149	2.324		5.538	2.599	
	Total	4.520	4.961	-9.76	5.735	5.244	8.56	7.137	5.523	22.62
	P0		0.636			0.920			1.190	
	P1		0.662			0.948			1.248	
	P2		0.668			0.953			1.215	
	P3		0.639			0.911			1.171	
PGF	Càrrega	1.634	1.587		1.730	1.589		1.632	1.633	
	Distribució	0.000	1.335		0.000	1.335		0.000	1.335	
	Filtratge	3.529	2.178		4.586	2.402		5.344	2.623	
	Total	5.163	5.099	1.24	6.317	5.326	15.67	6.976	5.591	19.86
	P0		0.776			1.000			1.220	
	P1		0.797			1.031			1.246	
	P2		0.807			1.019			1.261	
	P3		0.777			0.992			1.201	
VMF	Càrrega	1.678	1.737		1.606	1.952		1.784	1.715	
	Distribució	0.000	1.325		0.000	1.324		0.000	1.323	
	Filtratge	87.477	21.644		88.193	22.070		87.952	22.240	
	Total	89.154	24.706	72.29	89.799	25.346	71.77	89.736	25.278	71.83
	P0		20.256			20.658			20.859	
	P1		20.390			20.338			20.455	
	P2		20.418			20.137			19.917	
	P3		20.304			19.991			20.027	

Taula 7.2. Resultats per a la imatge **Kiev**

Filtre		Intensitat de soroll impulsiu								
		10,00 %			20,00 %			30,00 %		
		Mono	Multi	%	Mono	Multi	%	Mono	Multi	%
FMNF	Càrrega	3.862	3.466		3.544	3.456		4.803	3.469	
	Distribució	0.000	2.921		0.000	2.944		0.000	2.921	
	Filtratge	5.702	4.299		7.545	4.676		8.027	4.824	
	Total	9.564	10.687	-11.74	11.089	11.085	0.04	12.830	11.205	12.67
	P0		1.236			1.610			1.762	
	P1		1.365			1.629			1.845	
	P2		1.377			1.704			1.934	
	P3		1.356			1.757			1.973	
IFPGF	Càrrega	3.707	3.471		3.861	3.464		3.603	3.464	
	Distribució	0.000	2.917		0.000	2.917		0.000	2.225	
	Filtratge	8.580	4.998		11.354	5.552		13.962	5.449	
	Total	12.287	11.385	7.34	15.216	11.932	21.58	17.565	11.138	36.59
	P0		1.930			2.488			3.089	
	P1		1.906			2.554			3.250	
	P2		1.954			2.628			3.248	
	P3		1.968			2.615			3.180	
PGF	Càrrega	4.459	3.469		4.760	3.461		3.472	3.792	
	Distribució	0.000	2.917		0.000	2.919		0.000	2.230	
	Filtratge	9.660	5.146		11.732	5.655		13.786	5.501	
	Total	14.119	11.532	18.32	16.491	12.034	27.03	17.258	11.523	33.23
	P0		2.079			2.591			3.139	
	P1		2.184			2.706			3.276	
	P2		2.216			2.746			3.276	
	P3		2.230			2.770			3.228	
VMF	Càrrega	3.466	3.468		3.479	3.461		3.474	3.653	
	Distribució	0.000	2.895		0.000	2.894		0.000	2.207	
	Filtratge	187.02	46.755		190.51	46.782		189.26	46.647	
	Total	190.48	53.569	71.88	193.99	52.904	72.73	192.73	52.069	72.98
	P0		44.185			43.526			43.886	
	P1		43.373			44.262			43.400	
	P2		42.952			43.838			43.779	
	P3		43.515			43.763			43.179	

A la vista de les taules de resultats, podem comentar que en el filtratge d'una fase (VMF) el guany és més que considerable, obtenint-se millores al voltant del 72%, és a dir com que l'algorisme en un processador és de per sí lent, es veu immediatament la millora en les prestacions del filtratge paral·lel. Podem observar a les taules les millores substancials que oscil·len entre un 71-73 % de guany de velocitat depenent de cada imatge i intensitat de soroll.

Taula 7.3. Resultats per a la imatge **Formentera**

Filtre		Intensitat de soroll impulsiu								
		10,00 %			20,00 %			30,00 %		
		Mono	Multi	%	Mono	Multi	%	Mono	Multi	%
FMNF	Càrrega	12.762	14.100		13.116	12.564		12.175	12.136	
	Distribució	0.000	10.124		0.000	10.144		0.000	10.160	
	Filtratge	15.154	13.701		17.893	14.431		20.482	15.218	
	Total	27.916	37.925	-35.85	31.009	37.139	-19.77	32.656	37.514	-14.87
	P0		3.091			3.751			4.456	
	P1		3.067			3.662			4.267	
	P2		3.343			4.034			4.660	
	P3		3.827			4.637			5.475	
IFPGF	Càrrega	13.181	14.154		12.086	12.478		29.025	12.424	
	Distribució	0.000	10.687		0.000	10.169		0.000	10.219	
	Filtratge	21.943	16.150		29.320	16.670		36.688	18.079	
	Total	35.124	40.991	-16.70	41.406	39.320	5.05	65.713	40.722	38.03
	P0		4.404			5.949			7.399	
	P1		4.277			5.655			7.179	
	P2		4.572			6.503			8.005	
	P3		5.530			7.698			9.728	
PGF	Càrrega	15.884	12.137		13.776	12.145		13.287	12.258	
	Distribució	0.000	10.212		0.000	10.263		0.000	10.163	
	Filtratge	25.824	16.007		34.724	17.224		36.803	18.549	
	Total	41.708	38.357	8.04	48.500	39.632	18.29	50.091	40.971	18.21
	P0		5.253			6.527			7.858	
	P1		5.073			6.296			7.570	
	P2		5.523			6.948			8.608	
	P3		6.389			8.152			9.817	
VMF	Càrrega	12.650	12.629		13.385	12.301		12.790	16.439	
	Distribució	0.000	10.110		0.000	10.087		0.000	10.108	
	Filtratge	641.19	158.39		657.02	163.44		654.18	161.69	
	Total	653.89	181.13	72.30	670.40	185.83	72.28	666.97	188.24	71.78
	P0		147.72			152.81			151.07	
	P1		145.99			149.98			151.22	
	P2		150.43			150.00			154.56	
	P3		149.75			152.91			153.66	

Als filtratges de dos fases, com que són per naturalesa ràpids, el rendiment obtingut no és tan notable com els anteriors. De fet, pot observar-se que el rendiment del filtratge paral·lel no millora en certes condicions en alguns dels cassos/algorithmes analitzats. Vegem un estudi per separat dels filtres implementats:

FMNF Dels filtres implementats, aquest és el més ràpid de tots. Pot observar-se que en intensitats baixes de soroll i en les imatges més menudes és més competitiu el filtratge monoprocés, ja que el temps invertit en distribuir la

Taula 7.4. Resultats per al filtre FMNF en majors intensitats de soroll

Bee	Intensitat de soroll impulsiu								
	40,00 %			50,00 %			60,00 %		
	Mono	Multi	%	Mono	Multi	%	Mono	Multi	%
Càrrega	1.607	1.643		1.630	1.725		1.626	1.684	
Distribució	0.000	1.367		0.000	1.386		0.000	1.414	
Filtratge	3.642	2.218		4.086	2.347		4.460	2.413	
Total	5.231	5.228	0.05%	5.716	5.458	4.51%	6.086	5.511	9.45%
P0		0.800			0.901			0.981	
P1		0.818			0.936			1.049	
P2		0.839			0.951			1.041	
P3		0.774			0.898			0.983	

Kiev	Intensitat de soroll impulsiu								
	40,00 %			50,00 %			60,00 %		
	Mono	Multi	%	Mono	Multi	%	Mono	Multi	%
Càrrega	3.483	3.713		3.547	3.666		3.559	3.603	
Distribució	0.000	3.024		0.000	3.114		0.000	2.992	
Filtratge	10.073	5.371		11.026	5.786		12.009	5.904	
Total	13.556	12.108	10.68%	14.573	12.566	13.73%	15.568	12.499	19.71%
P0		2.198			2.481			2.769	
P1		2.341			2.680			2.859	
P2		2.529			2.611			2.929	
P3		2.510			2.757			2.891	

Formentera	Intensitat de soroll impulsiu								
	40,00 %			50,00 %			60,00 %		
	Mono	Multi	%	Mono	Multi	%	Mono	Multi	%
Càrrega	12.762	14.100		13.116	12.564		12.175	12.136	
Distribució	0.000	10.124		0.000	10.144		0.000	10.160	
Filtratge	15.154	13.701		17.893	14.431		20.482	15.218	
Total	39.818	37.925	4.75%	47.669	41.412	15.11%	52.901	42.898	23.32%
P0		5.095			5.848			6.330	
P1		4.938			5.660			6.000	
P2		5.863			6.308			7.112	
P3		6.707			7.459			8.404	

imatge fa que es ralentisca el rendiment multiprocessador. Pot observar-se a les taules de resultats que a mesura que augmenta la intensitat de soroll el rendiment del filtratge paral·lel és cada cop millor, i això ens ha portat a fer experimentacions d'aquest filtre amb majors intensitats de soroll. A la taula 7.4 es presenta un estudi amb les tres imatges estudiades amb intensitats superiors de soroll (contaminades amb un 40, 50 i 60 %). A causa que sota aquestes condicions de soroll es requereix més potència de procés, ara sí s'observen guanys que varien entre un 10%, 19% i 24% en cada imatge. Noteu també que en la imatge Formentera, amb intensitats baixes de soroll la versió monoprocessador és molt més ràpida. Això és a

causa que aquesta imatge presenta zones molt homogènies i molt grans (les zones de la mar i el llac) i això fa que el filtre FMNF siga molt ràpid. Amb tot això podem concloure que per al filtre FMNF els resultats obtinguts han sigut molt bons, però que aquesta millora es fa visible a l'augmentar tant la talla de la imatge com el nivell de contaminació de la mateixa.

IFPGF Aquest filtre és més lent que el FMNF, però bastant ràpid també.

Recordem que la característica principal del filtre és la qualitat de les imatges obtingudes, tot i això amb un rendiment computacional més que suficient. Els resultats obtinguts en aquest filtre confirmen el que hem estat comentant, quan la intensitat de soroll comença a augmentar el rendiment del filtratge paral·lel comença a millorar de manera significativa.

PGF En aquest tercer filtre de la comparativa el guany és clar, ja que és el més lent dels analitzats de dues fases. Això ho podem deduir ja amb intensitat de soroll més baixes on el filtratge paral·lel ja millora el filtre PGF i amb intensitats més altes s'arriba a aconseguir guanys computacionals de fins un 33%.

7.6 Conclusions i Treball Futur

Per enllestir el treball, cal comentar que els resultats obtinguts són molt positius, com s'ha pogut observar a les taules comparatives. Tot i això hem de fer una anàlisi més detallada del problema quan la càrrega de treball no és alta.

Les tècniques de processat paral·lel amb MPI són més eficients si el volum d'informació a transferir és relativament menut i la quantitat de procés és alta. En aquestes implementacions de filtratge d'imatges, independentment de la potència de còmput necessària per a filtrar les imatges, hi ha un gran volum de dades a distribuir. Tal i com s'ha fet la implementació i experimentació, el procés pare (procés 0) s'encarrega de dividir la imatge en 4 trossos, un per a ell i la resta per als restants processos. Posteriorment la imatge s'ha d'enviar a la resta de processos, i en el cas de Formentera un quart d'imatge ocupa al voltant de $30MB$. Això ens implica que intensitats de soroll baixes (que requereixen menys computacions) apareix un coll de botella a la distribució de les dades, la qual cosa provoca que les tècniques de filtratge paral·lel en aquests casos no siguin convenients.

Si d'alguna manera aconseguíem reduir la distribució de les imatges els resultats haurien sigut encara més positius, i ací apareix un possible treball futur. Una possibilitat serà fer que el procés 0 (o *pare*) distribuïska a cada subprocés solament la informació de quina porció d'imatge ha de filtrar (és a dir sols les coordenades de la imatge que li correspon filtrar) i cada procés carrega la seua porció d'imatge, la filtre i la dese a la imatge final. Possiblement encara no seria suficient amb això, ja que ocorreria que els processos

farien cua al disc per llegir/escriure la imatge, traslladant el coll de botella al disc. La solució seria aplicar alguna tècnica de càrrega distribuïda d'arxius.

Per concloure cal dir que aquestes novedoses implementacions de filtratge d'imatges en paral·lel són convenientes per imatges grans en nivells alts de soroll. Com és evident el tipus d'aplicacions a les que estarien destinats aquests filtres serien el procés d'imatges de satèl·lit, imatges mèdiques d'altíssima resolució i similars, en entitats mèdiques, governamentals i militars on el factor temps és clau i es poden disposar de clusters de computadors per poder executar aquestes implementacions.

Part III

Conclusions i Treball Futur

8 Conclusions i Treball Futur

En aquesta Tesi Doctoral, s'han dissenyat diverses tècniques de filtrat basades en l'ús de una mètrica fuzzy (en el sentit de George i Veeramani). Com s'ha vist, les tècniques dissenyades s'aprofiten de les mètriques fuzzy així com de les tècniques de *peer group* per dissenyar tècniques robustes de filtrat d'imatges en color. A banda de les conclusions estudiades per separat en cadascun dels Capítols, podem remarcar les següents conclusions generals:

- La mètrica fuzzy M introduïda a la Secció 3.2 al Capítol 4 és adequada per al processament d'imatges a l'espai de color RGB, tenint com avantatge addicional el menor cost computacional d'aquesta última.
- Les tècniques de *peer group*, combinades amb les mètriques fuzzy són un mecanisme ràpid en la detecció de soroll impulsiu.
- Si combinem una tècnica ràpida de detecció de soroll amb un mecanisme eficient de substitució de píxels sorollosos s'aconsegueixen tècniques de filtrat molt competitives davant les tècniques de l'estat de l'art actual.
- De cara al procés d'imatges grans, podem recórrer a tècniques d'implementació amb computació paral·lela

Com a conseqüència de les conclusions exposades anteriorment, les futures línies d'investigació i possibles àrees de recerca apunten a:

- Estudiar amb més profunditat l'aplicabilitat de les tècniques fuzzy per a processar imatges multicanal tenint en compte la correlació entre els canals de la imatge de mode distint a les aproximacions vectorials.
- Investigar possibles aplicacions de les mètriques fuzzy i de les eines de la lògica fuzzy per a la reducció d'altres defectes en imatges en color (no solament soroll). Per exemple, un tipus de defectes recents que apareixen en imatges en color de càmeres i escàners són els anomenats píxels calents i píxels morts (més informació a www.netimage.com)
- Aplicar les tècniques anteriors a altres camps de processament de imatges, com per exemple entrellaçat i desentrellaçat de vídeo, segmentació, detecció de vores, etc.
- Aplicar les tècniques anteriors fora d'imatges de consum, com pugen ser imatges mèdiques, tomografies, radiografies, cadenes de ADN, etc.

Referències

1. H. Allende, J. Galbiati, A non-parametric filter for image restoration using cluster analysis, *Pattern Recognition Letters* 25 8 (2004) 841-847.
2. K. Arakawa, Median filter based on fuzzy rules and its application to image restoration, *Fuzzy Sets and Systems*, 77 1 (1996) 3-13.
3. G. Artico and R. Moresco, On fuzzy metrizable, *J. Math. Appl.* **107** (1985) 144-147.
4. J. Astola, P. Haavisto, Y. Neuvo, Vector Median Filters, *Proc. IEEE.* 78 4 (1990) 678-689.
5. K.E. Barner, T.C. Aysal, Polynomial weighted median filtering, *IEEE Transactions on Signal Processing*, 54 2 (2006) 636-650.
6. V. Barnett, The ordering of multivariate data. *Journal of royal statistical society A* 139 2 (1976) 331-354.
7. V. Barnett, T. Lewis, *Outliers in multivariate data*, John Wiley and Sons, New York 1994.
8. M. Barni, F. Buti, F. Bartolini, V. Capellini, A Quasi-Euclidean Norm to Speed Up Vector Median Filtering, *IEEE Transactions on Image Processing* 9 10 (2000) 1704-1709.
9. M. Barni, A Fast Algorithm for 1-Norm Vector Median Filtering, *IEEE Transactions on Image Processing* 6 10 (1997) 1452-1455.
10. G. Peris-Fajarnés, B. Roig and A. Vidal, "Rank-Ordered Differences Statistic Based Switching Vector Filter", *Lecture Notes in Computer Science* vol. 1, 74-81 (2006). Proceedings of the *International Conference on Image Analysis and Recognition ICIAR 2006*
11. J. Camacho, S. Morillas, P. Latorre, Efficient impulsive noise suppression based on statistical confidence limits, *Journal of Imaging Science and Technology* 5 6 (2006) 427-436.
12. Joan-Gerard Camarena, V. Gregori, S. Morillas, A. Sapena, "Fast detection and removal of impulsive noise using peer groups and fuzzy metrics", *Journal of Visual Communication and Image Representation*, 19 (2008) 20-29.
13. Joan-Gerard Camarena, Valentín Gregori, Samuel Morillas and Guillermo Peris-Fajarnés "New Method for Fast Detection and Removal of Impulsive Noise Using Fuzzy Metrics", *Lecture Notes in Computer Science* vol. 1, pp. 359-369 (2006). Proceedings of the *International Conference on Image Analysis and Recognition ICIAR 2006*
14. Joan-Gerard Camarena, Valentín Gregori, Samuel Morillas and Almanzor Sapena "Some Improvements for image filtering using peer group techniques". *Enviat a Image and Vision Computing*.

15. G. L. Cain Jr. and R. H. Kasriel, Fixed and periodic points of local contraction mappings on probabilistic metric spaces, *Mathematical Systems Theory* **9** (1976) 289-297.
16. M.E. Celebi, H.A. Kingravi, Y.A. Aslandogan, Nonlinear vector filtering for impulsive noise removal from color images, *Journal of Electronic Imaging* **16** **3** (2007) 1-21.
17. V. Chatzis, I. Pitas, Fuzzy scalar and vector median filters based on fuzzy distances, *IEEE Transactions on Image Processing* **8** 5 (1999) 731-734.
18. C.H. Chen, *Fuzzy logic and neural network handbook*, McGraw-Hill, New York 1996.
19. C. L. Chang, Fuzzy topological spaces, *J. Math. Anal. Appl.* **24** (1968) 182-190.
20. H.A. David, *Order Statistics*. John Wiley and Sons, New York 1981.
21. Y. Deng, C. Kenney, MS Moore, BS Manjunath, Peer group filtering and perceptual color image quantization, *Proceedings of IEEE international symposium on circuits and systems* **4** (1999) 21-4.
22. Deng Zi-ke, Fuzzy pseudo metric spaces, *J. Math. Anal. Appl.* **86** (1982) 74-95.
23. Deng Zi-ke, Separation axioms for completeness and total boundedness in fuzzy pseudometric spaces, *J. Math. Anal. Appl.* **112** (1985) 141-150.
24. M. Elad, *On the origin of bilateral filter and ways to improve it*, IEEE Transactions on Image Processing **11** 10 (2002) 1141-1151.
25. R. Engelking, *General Topology*, PWN-Polish Sci. Publ. Warsawa (1977) Warsaw, Poland.
26. M. A. Erceg, Metric spaces in fuzzy set theory, *J. Math. Anal. Appl.* **69** (1979) 205-230.
27. F. Farbiz and M. B. Menhaj, A fuzzy logic control based approach for image filtering, in *Fuzzy Techniques in Image Processing*, 1st ed., vol. 52, E.E. Kerre and M. Nachtgael, Eds. Heidelberg: Physica Verlag, 2000, pp. 194-221.
28. F.J. Gallegos-Funes, V. Ponomaryov, Real-time image filtering scheme based on robust estimators in presence of impulse noise, *Real-Time Imaging* **10** **2** (2004) 69-80.
29. R. Garnett, T. Huegerich, C. Chui, W. He, *A universal noise removal algorithm with an impulse detector*, IEEE Transactions on Image Processing **14** 11 (2005) 1747-1754.
30. A. George and P. Veeramani, On Some results in fuzzy metric spaces, *Fuzzy Sets and Systems* **64** **3** (1994) 395-399.
31. A. George and P. Veeramani, Some theorems in fuzzy metric spaces, *Journal of Fuzzy Mathematics* **3** (1995) 933-940.
32. J. Goguen, L-fuzzy sets, *J. Math. Anal. Appl.* **18** (1967) 145-174.
33. V. Gregori and S. Romaguera, Some properties of fuzzy metric spaces, *Fuzzy Sets and Systems* **115** (2000) 485-489.
34. V. Gregori, S. Romaguera, Characterizing completable fuzzy metric spaces, *Fuzzy Sets and Systems* **144** **3** (2004) 411-420.
35. M. Grabiec, Fixed points in fuzzy metric spaces, *Fuzzy Sets and Systems* **27** (1988) 385-389.
36. A. George and P. Veeramani, On some results of analysis for fuzzy metric spaces, *Fuzzy Sets and Systems* **90** (1997) 365-368.
37. G. Hewer, C. Kenney, L. Peterson, A. Van Nevel, Applied partial differential variational techniques, *Proceedings of International Conference on Image Processing ICIP'97*, **3** (1997) 372-375.

38. J. Y. F. Ho, Peer region determination based impulsive noise detection, *Proceedings of International Conference on Acoustics, Speech and Signal Processing ICASSP'03* 3 (2003) 713-716.
39. S. Hore, B. Qiu, and H.R. Wu, Improved vector filtering for color images using fuzzy noise detection, *Optical Engineering* , 42 6 (2003) 1656-1664.
40. P.S. Huber, *Robust Statistics* (John Wiley and Sons, New York, 1981)
41. H. Hu, Fuzzy topological spaces, *J. Math. Anal. Appl.* **110** (1985) 141-178.
42. B. Hutton, Products of fuzzy topological spaces, *Topology Appl.* **11** (1980) 59-67.
43. I. Jermolaeva, On a Hausdorffness function of a fuzzy topological space, *Zbornik Radova Filozof. Fakulteta u Nižu. Ser. Mat.* **2** (1988) 73.
44. L. Jin, D. Li, A switching vector median filter based on the CIELAB color space for color image restoration, *Signal Processing* **87** **6** (2007) 1345-1354.
45. L. Jin, D. Li, An efficient color impulse detector and its application to color images, *IEEE Signal Processing Letters*, **14** **6** (2007) 397-400.
46. I. Kalaykov and G. Tolt, Real-time image noise cancellation based on fuzzy similarity, in *Fuzzy Filters for Image Processing*, 1st ed., vol. 122, M. Nachttegael, D. Van der Weken, D. Van De Ville and E. E. Kerre, Eds. Heidelberg: Physica Verlag, 2003, pp. 54-71.
47. D.G. Karakos, P.E. Trahanias, Generalized multichannel image-filtering structure, *IEEE Transactions on Image Processing* 6 7 (1997) 1038-1045.
48. O. Kaleva and S. Seikkala, On fuzzy metric spaces, *Fuzzy Sets and Systems* **12** (1984) 215-229.
49. C. Kenney, Y. Deng, BS Manjunath, G. Hower, Peer group image enhancement, *IEEE Transactions on Image Processing* 10 2 (2001) 326-334.
50. M. Nachttegael, D. Van der Weken, D. Van De Ville, E.E. Kerre (eds.): *Fuzzy Filters for Image Processing*, Vol. 122 Springer Physica Verlag, Berlin Heidelberg New York, 2003.
51. E.E. Kerre, *Fuzzy sets and approximate Reasoning* Xian Jiaotong University Press, 1998.
52. L. Khriji, M. Gabbouj, Vector median-rational hybrid filters for multichannel image processing, *IEEE Signal Processing Letters*, 6 7 (1999) 186-190.
53. L. Khriji, M. Gabbouj, Adaptive fuzzy order statistics-rational hybrid filters for color image processing, *Fuzzy Sets and Systems*, 128 1 (2002) 35-46.
54. KODAK test images database, <http://r0k.us/graphics/kodak/>.
55. I. Kramosil and J. Michalek, Fuzzy metric and statistical metric spaces, *Kybernetika* **11** (1975) 326-334.
56. Y. Li, J. Bacca-Rodríguez, G.R. Arce, Weighted median filters for multichannel signals, in *Proc. Int. Conf. Acoustics, Speech and Signal Processing ICASSP'05* vol. IV, (2005) 157-160.
57. R. Lowen, Fuzzy topological spaces and fuzzy compactness, *J. Math. Anal. Appl.* **56** (1976) 621-633.
58. R. Lowen, Initial and final fuzzy topologies and the fuzzy Tychonoff theorem, *J. Math. Anal. Appl.* **58** (1977) 11-21.
59. L. Lucat, P. Siohan, D. Barbac, Adaptive and global optimization methods for weighted vector median filters, *Signal Processing: Image Communication* **17** **7** (2002) 509-524.
60. L. Lucchese, S.K. Mitra, A new class of chromatic filters for color image processing: Theory and applications, *IEEE Transactions on Image Processing* , 14 4 (2004) 534-548.

61. R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Selection weighted vector directional filters, *Computer Vision and Image Understanding*, 94 1-3 (2004) 140-167 2004.
62. R. Lukac, Adaptive vector median filtering, *Pattern Recognition Letters* 24 12 (2003) 1889-1899.
63. R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, A.N. Venetsanopoulos, Vector Filtering for Color Imaging, *IEEE Signal Processing Magazine, Special Issue on Color Image Processing* 22 1 (2005) 74-86.
64. R. Lukac, K.N. Plataniotis, B. Smolka, A.N. Venetsanopoulos, cDNA Microarray Image Processing Using Fuzzy Vector Filtering Framework, *Fuzzy Sets and Systems: Special Issue on Fuzzy Sets and Systems in Bioinformatics*, 152 1 (2005) 17-35.
65. R. Lukac, K.N. Plataniotis, B. Smolka, A.N. Venetsanopoulos, A Multichannel Order-Statistic technique for cDNA Microarray Image Processing, *IEEE Transactions on Nanobioscience* 3 4 (2004) 272-285.
66. R. Lukac, K.N. Plataniotis, B. Smolka, A.N. Venetsanopoulos, Generalized Selection Weighted Vector Filters, *EURASIP Journal on applied signal processing: Special Issue on Nonlinear signal and image processing*, 2004 12 (2004) 1870-1885.
67. R. Lukac, Adaptive Color Image Filtering Based on Center Weighted Vector Directional Filters, *Multidimensional Systems and Signal Processing* 15 2 (2004) 169-196.
68. R. Lukac, K.N. Plataniotis, A.N. Venetsanopoulos, B. Smolka, A Statistically-Switched Adaptive Vector Median Filter, *Journal of Intelligent and Robotic Systems* 42 4 (2005) 361-391.
69. R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Vector sigma filters for noise detection and removal in color images, *Journal of Visual Communication and Image Representation* 17 1 (2006) 1-26.
70. R. Lukac, K.N. Plataniotis, A.N. Venetsanopoulos, Color image image denoising using evolutionary computation, *International Journal of Imaging Systems and Technology* 15 5 (2005) 236-251 .
71. R. Lukac, K.N. Plataniotis, A taxonomy of color image filtering and enhancement solutions, in *Advances in Imaging and Electron Physics*, (eds.) P.W. Hawkes, Elsevier, 140 (2006) 187-264.
72. R. Lukac and K.N. Plataniotis, *Color Image Processing: Methods and Applications*. Boca Raton, FL., CRC Press / Taylor & Francis, 2006.
73. Z. Ma, D. Feng, H.R. Wu, A neighborhood evaluated adaptive vector filter for suppression of impulsive noise in color images, *Real-Time Imaging*, 11 5-6 (2005) 403-416.
74. Z. Ma, H.R. Wu, D. Feng, Partition-based vector filtering technique for suppression of noise in digital color images, *IEEE Transactions on Image Processing* , 15 8 (2006) 2324-2342.
75. Z. Ma, H.R. Wu, B. Qiu, A robust structure-adaptive vector filter for color image restoration, *IEEE Transactions on Image Processing* , 14 12 (2005) 1990-2001.
76. Z. Ma, H. R. Wu, B. Qiu, A window adaptive hybrid vector filter for color image restoration, *Proceedings of International Conference on Acoustics, Speech and Signal Processing ICASSP'04* 3 (2004) 205-208.
77. D.L. MacAdam, *Visual sensitivities to color differences in daylight*, J. Opt. Soc. Am., 33 (1942) 247-274.

78. K. Menger, Statistical metrics, *Proc. Nat. Acad. Sci.* **28** (1942) 535-537.
79. P. Pacheco, Parallel Programming with MPI, Morgan Kaufmann Publishers Inc, 978-1-55860-339-4 , 1996
80. E. Pap and O. Hadzic and R. Mesiar, A fixed point theorem in probabilistic metric spaces and an application, *J. Math. Anal. Appl.* **202** 433-449.
81. E. Parau and V. Radu, Some remarks on Tardiff's fixed point theorem on Menger spaces, *Portugal. Math.* **54** (1997) 431-440.
82. P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12** 5 (1990) 629-639.
83. I. Pitas, A.N. Venetsanopoulos, *Nonlinear digital filters: Principles and Applications*, Kluwer Academic Publishers, Boston, 1990.
84. K.N. Plataniotis, A.N. Venetsanopoulos, *Color Image processing and applications*, Springer-Verlag, Berlin, 2000.
85. V.I. Ponomaryov, F.J. Gallegos-Funes, A. Rosales-Silva, Real-Time color imaging based on RM-filters for impulsive noise reduction, *Journal of Imaging Science and Technology* **49** 3 (2005) 205-219.
86. V. Rafu, Some fixed point theorems in probabilistic metric spaces, stability problems for stochastic models, *Lecture Notes in Mathematics* **1233** (1987) 125-133.
87. S. Morillas, V. Gregori, G. Peris-Fajarnés, P. Latorre, "A new vector median filter based on fuzzy metrics", *ICIAR 2005, Lecture Notes in Computer Science* **3656** (2005) 81-90.
88. S. Morillas, V. Gregori, G. Peris-Fajarnés, P. Latorre, A fast impulsive noise color image filter using fuzzy metrics, *Real-Time Imaging* **11** 5-6 (2005) 417-428.
89. S. Morillas, V. Gregori, G. Peris-Fajarnés, Isolating impulsive noise pixels in color images by peer group techniques, *Computer Vision and Image Understanding*, **110** 1 (2008) 102-116.
90. S. Morillas, V. Gregori, G. Peris-Fajarnés, A. Sapena, New Adaptive Vector Filter Using Fuzzy Metrics, *Journal of Electronic Imaging* **16** 3 (2007) 033007:1-15.
91. S. Morillas, V. Gregori, G. Peris-Fajarnés, A. Sapena, Local Self-Adaptive Fuzzy Filter for Impulsive Noise Removal in Color Images, *Signal Processing*, **88** 2 (2008) 390-398.
92. S. Morilla, V. Gregori, G. Peris-Fajarnés, P. Latorre, A Fast Impulsive Noise Color Image Filtering Using Fuzzy Metrics, *Real-Time Imaging* **11** 5-6 (2005) 417-428.
93. A. Sapena, A contribution to the study of fuzzy metric spaces, *Appl. Gen. Topology* **2** 1 (2001) 63-76.
94. S. Schulte, S. Morillas, V. Gregori, E.E. Kerre, A New Fuzzy Color Correlated Impulsive Noise Reduction Method, *IEEE Transactions on Image Processing* **16** 10 (2007) 2565-2575.
95. B. Schweizer and A. Sklar, Statistical metric spaces, *Pacific. J. Math.* **10** (1960) 314-334.
96. B. Schweizer and A. Sklar, Probabilistic Metric Spaces, *Elsevier Science Publishing Co.* (1983) New York, USA.
97. Y. Shen, K. Barner, *Fuzzy vector median-based surface smoothing*, *IEEE Transactions on Visualization and Computer Graphics* **10** 3 (2004) 252-265.

98. Y. Shen, K.E. Barner, Marginal fuzzy median and fuzzy vector median filtering of color images, in *Proc. 37th Annual Conf. Inf. Sciences & Systems* (2003).
99. Y. Shen, K.E. Barner, Optimization of fuzzy vector median filters, in *Proc. 38th Annual Conf. Inf. Sciences & Systems* (2004).
100. A. P. Šostak, Two decades of fuzzy topology: basic ideas, notions and results, *Russian Math. Surveys* **4** **6** (1989) 125-186.
101. B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, Similarity based impulsive noise removal in color images, *International Conference on Image Processing ICIP 2003*.
102. B. Smolka, R. Lukac, A. Chydzinski, K.N. Plataniotis, W. Wojciechowski, Fast adaptive similarity based impulsive noise reduction filter, *Real-Time Imaging, Special Issue on Spectral Imaging* **9** **4** (2003) 261-276.
103. B. Smolka, A. Chydzinski, K.N. Plataniotis, A.N. Venetsanopoulos, New filtering technique for the impulsive noise removal in color images, *Mathematical Problems in Engineering* **1** (2004) 79-91.
104. B. Smolka, K.N. Plataniotis, A. Chydzinski, M. Szczepanski, A.N. Venetsanopoulos, K. Wojciechowski, Self-adaptive algorithm of impulsive noise reduction in color images, *Pattern Recognition* **35** **8** (2002) 1771-1784.
105. B. Smolka, A. Chydzinski, Fast detection and impulsive noise removal in color images, *Real-Time Imaging* **11** 5-6 (2005) 389-402.
106. B. Smolka, K.N. Plataniotis, "Ultrafast technique of impulsive noise removal with application to microarray image denoising", *Lecture Notes in Computer Science* **3656** (2005) 990-997.
107. S. Schulte, M. Nachtgael, V. De Witte, D. Van der Weken, E. E. Kerre, A Fuzzy Impulse Noise Detection and Reduction Method, *IEEE Transactions on Image Processing*, **15** **5** (2006) 1153-1162.
108. Schulte, S., De Witte, V., Nachtgael, M., Van der Weken, D., Kerre, E. E.: Fuzzy two-step filter for impulse noise reduction from color images. *IEEE Transactions on Image Processing* **15** **11** (2006) 3567-3578
109. S. Schulte, V. De Witte, M. Nachtgael, D. Van der Weken, E. E. Kerre, Fuzzy random impulse noise reduction method, *Fuzzy Sets and Systems*, **158** **3** (2007) 270-283.
110. M. Szczepanski, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, On the distance function approach to color image enhancement, *Discrete Applied Mathematics*, **139** 1-3 (2004) 283-305.
111. R. M. Tardiff, Contraction maps on probabilistic metric spaces, *J. Math. Anal. Appl.* **165** (1992) 517-523.
112. C. Tomasi, R. Manduchi, *Bilateral filter for gray and color images*, Proc. IEEE International Conference Computer Vision, 1998, 839-846.
113. P.E. Trahanias, A.N. Venetsanopoulos, Vector directional filters-a new class of multichannel image processing filters, *IEEE Trans. Image Process.* **2** **4** (1993) 528-534.
114. P.E. Trahanias, D. Karakos, A.N. Venetsanopoulos, Directional processing of color images: theory and experimental results, *IEEE Trans. Image Process.* **5** **6** (1996) 868-880.
115. E. Trillas, *Fundamentos e introducción a la ingeniería fuzzy*, Omron Electronics, Madrid 1994.
116. H.H. Tsai, P.T. Yu, Genetic-based fuzzy hybrid multichannel filters for color image restoration, *Fuzzy Sets and Systems*, **114** **2** (2000) 203-224.

117. T. Viero, K. Oistamo, Y. Neuvo, Three-dimensional median-related filters for color image sequence filtering, *IEEE Transactions on Circuits and Systems for Video Technology* 4 2 (1994) 129-142.
118. J. H. Wang and H. C. Chiu, An adaptive fuzzy filter for restoring highly corrupted images by histogram estimation, *Proceedings of the National Science Council -Part A* 23 (1999) 630-643.
119. J.H. Wang, W.J. Liu, L.D. Lin, Histogram-Based Fuzzy Filter for Image Restoration, *IEEE Transactions on Systems man and cybernetics part B-cybernetics*, 32 2 (2002) 230-238.
120. H. Xu, G. Zhu, H. Peng and D. Wang, Adaptive fuzzy switching filter for images corrupted by impulse noise, *Pattern Recognition Letters*, 25 (2004) 1657-1663.
121. J. Yen, R. Langari, *Fuzzy logic: intelligence, control and information*, Prentice-Hall, New Jersey 1998.
122. S.Q. Yuan, Y.H. Tan, Impulse noise removal by a global-local noise detector and adaptive median filter, *Signal Processing* **86** 8 (2006) 2123-2128.
123. L.A. Zadeh, Fuzzy sets, *Inform. Control* **8** (1965) 338-353.

