LUCRAREA NR. 3
FUNCTII SPECIALE SI TEHNICI DE PRELUCRARE AUTOMATA CU GIMP

Aplicatii web in GIMP

1. Obiective

Aceasta lucrare de laborator isi propune familiarizarea studentului cu aplicatiile web disponibile in programele de editare fotografica. Fie ca vorbim de reducerea dimensiunilor unei imagini pentru publicarea pe web, fie despre realizarea unei librarii de linkuri pe o imagine, toate abilitatile urmarite de aceasta lucrare au aplicabilitate larga in designul web.

2. Uneltele de culoare

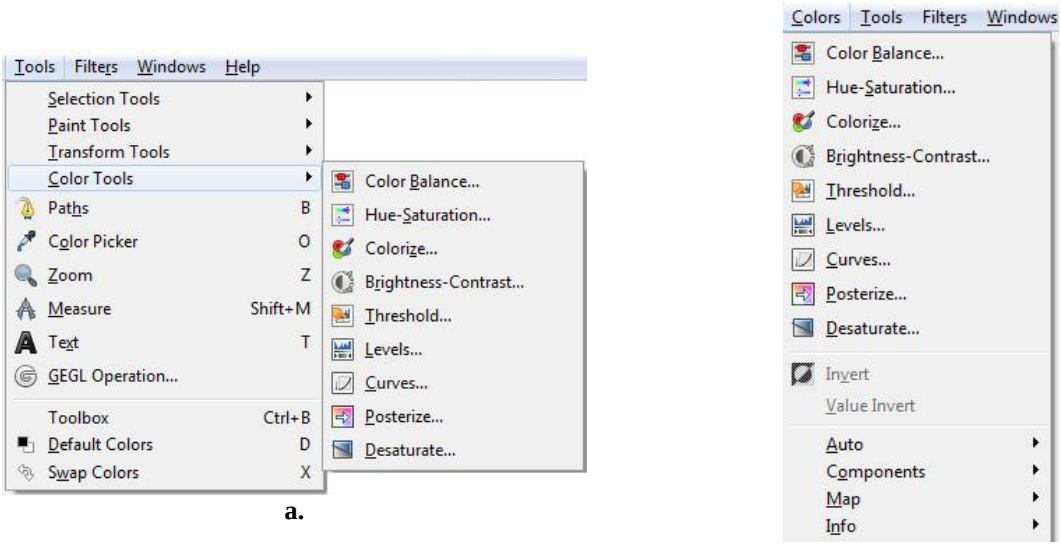
Am văzut în laboratorul precedent cum putem folosi în GIMP uneltele de selecție, uneltele pentru pictură și desen, unealta traseu și unealta text. Un alt set foarte important de unelte este grupul uneltelor de culoare. Spre deosebire de grupurile de unelte pe care le-am folosit în laboratorul precedent, acest grup de unelte grafice nu poate fi accesat direct din Toolbox. Il putem însă accesa prin cele două modalități descrise în Figura 2, fie prin intermediul meniului „*Tools* | *Color Tools*” (a), fie direct prin intermediul meniului „*Colors*” (b). Această secțiune va fi dedicată în continuare prezentării acestor unelte.

2.1. Balanța de culori (Color Balance)

Această unealtă este cel mai simplu accesată din meniul „*Colors* | *Color Balance*”. Odată deschis dialogul său de opțiuni, putem selecta căror pixeli să le fie modificată culoarea folosindune de opțiunea „*range to adjust*”. Putem în acest fel să modificăm pixelii cei mai închiși („*Shadows*”), pe cei medii („*Midtones*”) sau pe cei mai luminoși („*Highlights*”). Este în general indicat să nu modificăm cei mai deschiși sau cei mai închiși pixeli, pentru a nu schimba aspecul pixelilor albi și negri. In acest fel, deși putem modifica destul de mult pixelii de luminozități medii, putem păstra aspectul natural al fotografiei. In Figura 1 de exemplu am modificat imaginea originală „*Laborator GAC/loki.jpg*” (a) pentru a schimba culoarea predominantă a imaginii din verde în albastru (b), respectiv roșu (c). Ajustarea culorilor se realizează pe cele 3 axe de culori RGB, fiecare axă având la un capăt una dintre culorile roșu, verde sau albastru, iar la celălalt capăt complementele lor. Bifarea opțiunii „Preserve luminosity” va asigura păstrarea luminozității fiecărui pixel. Prin selectarea și deselectarea opțiunii „Preview” putem alterna între imaginea originală și imaginea modificată.

a. **b.** **c.**

Figura 1. Folosirea uneltei "Color Balance" pentru modificarea echilibrului culorilor: a. imaginea originală; b. imaginea deplasată spre albastru; c. imaginea deplasată spre roșu



a. **b.**

Figura 2. Accesarea uneltelor de culoare GIMP: a. prin intermediul meniului Tools; b. direct prin intermediul meniului Colors

2.2. Nuanță și Saturație (Hue-Saturation)

Această unealtă poate fi folosită pentru schimbarea nuanței culorilor. Spre deosebire de funcția „Color Balance" prezentetată precedent, selecția pixelilor a căror nuanță va fi schimbată este facută pe baza culorilor (prin intermediul opțiunii „Primary Color to Adjust"), nu a luminozității. Cercul culorilor pe baza căruia vom face selecția pixelilor este alcătuit din 6 zone, fiecare zonă corespunzând uneia din culorile de bază (R – roșu, G – verde, B – albastru) sau uneia din culorile lor complementare (C – azuriu, M – purpuriu, Y – galben).

După selecția culorii a fi schimbate, vom opera modificările prin deplasarea cursorului „*Hue*". Modificarea este exprimată în grade, iar aceasta reprezentă unghiul deplasării culorii selectate pe cercul culorilor. Atunci când unghiul are valoare pozitivă, deplasarea se face în sens invers acelor de ceasonic, iar când unghiul are valoare negativă, în sens direct. Rezultă așadar că

atât pentru o rotație maximă de 180° cât și pentru una inversă de -180°, culoarea selectată se va transforma în complementul ei.

Prin deplasarea cursorului „*Lightness*” putem micșora sau mări luminozitatea pixelilor selectați, iar prin deplasarea cursorului „*Saturation*” putem regla intensitatea culorii. Pentru o saturație de 100 culoarea selectată va avea o intensitate maximă, iar pentru o saturație de -100, pixelii selectați vor fi reprezentați doar în nuanțe de gri.

Folosind așadar doar opțiunile disponibile în unealta „Hue-Saturation” putem recrea efectul obținut anterior folosind unealta „Color-Balance”. Noul set de imagini este reprezentat în Figura 3.

2.3. Niveluri (Levels)

Această unealtă poate fi folosită pentru a închide sau deschide o imagine, pentru schimbarea contrastului sau pentru corecția culorilor. Activarea acestei unelte se face din meniul „*Tools* | *Color Tools*” (Figura 2a) sau din meniul „*Colors*” (Figura 2b). Imediat după activare vom putea observa o fereastră de dialog precum cea din Figura 4.

a. **b.** **c.**

Figura 3. Folosirea uneltei “Hue-Saturation” pentru modificarea echilibrului culorilor: a. imaginea originală; b. imaginea deplasată spre albastru; c. imaginea deplasată spre roșu

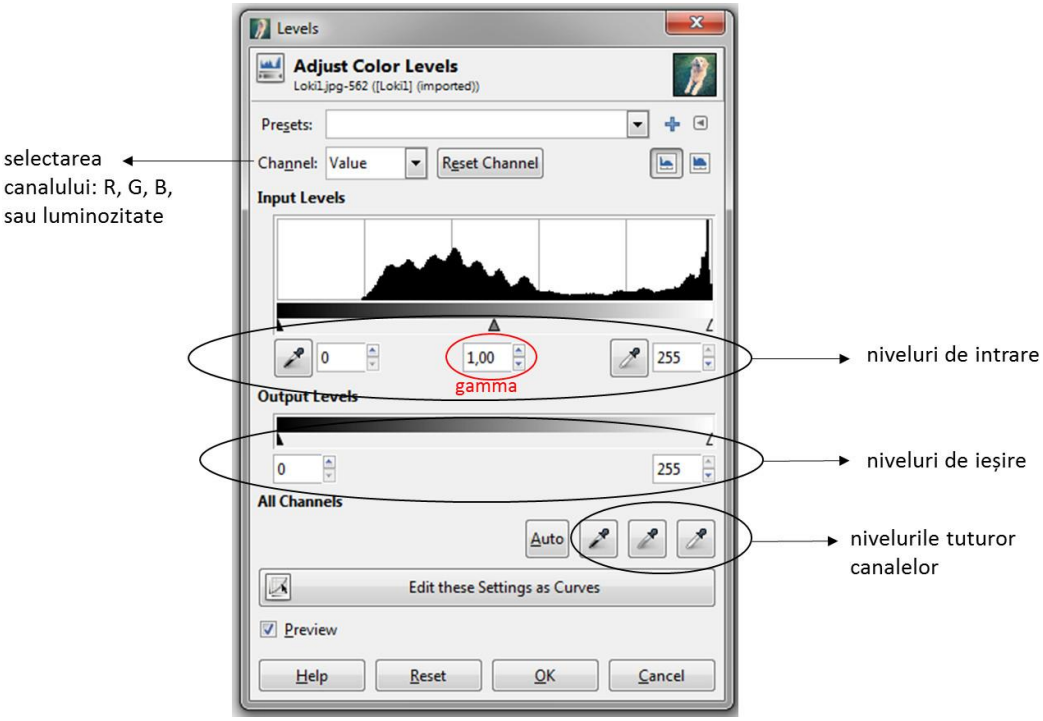


Figura 4. Dialogul folosit pentru reglarea uneltei „Niveluri”(Levels)

Modificările pe care le operăm cu această unealtă se vor face în general separat pentru fiecare canal în parte. Primul lucru pe care îl vom face așadar după activarea uneltei este să selectăm canalul pe care îl dorim modificat, folosind meniul derulant „Channel” din Figura 4. Histograma canalului selectat este reprezentată imediat sub meniul derulant de selecție.

Nivelurile de intrare pot fi reglate în trei moduri: fie prin intermediul triunghiurilor mobile de sub histogramă, fie prin intermediul valorilor numerice aflate în căsuțele de sub histogramă, fie prin intermediul uneltelor de extracție a culorilor. Nivelurile de ieșire pot fi reglate doar în două moduri: folosind triunghiurile mobile și folosind căsuțele numerice din dreptul acestora.

După selecția limitelor nivelurilor de intrare și de ieșire, această unealtă va transforma valoarea fiecărui pixel având un anumit nivel de intrare în nivelul de ieșire proporțional corespunzător. De exemplu, să presupunem că selectăm canalul roșu (red). La intrare alegem limita inferioară la 30 și limita superioară la 200, iar la ieșire fixăm limita inferioară la 0, iar limita superioară la 250. Atunci toți pixelii care aveau inițial valoarea canalului roșu inferioară lui 30 vor avea după transformare o valoare a canalului roșu nulă, iar toți pixelii care erau superiori lui 200 vor avea o valoare a canalului roșu de 250. Toate valorile de roșu intermediare, cuprinse între 30 și 200, se vor transforma, după o lege liniară, în valori de roșu în intervalul 0 – 250. Transformarea valorilor între nivelurile de intrare și cele de ieșire se va face după o lege liniară doar dacă parametrul gamma (parametrul notat cu roșu în Figura 4) corespunzător nivelurilor de intrare este egal cu 1. In cazul că acest parametru este supraunitar, valorile pixelilor pe canalul ales vor fi mai mari decât pentru cazul liniar (imaginea se va lumina), iar în cazul invers aceste valori vor fi inferioare cazului liniar (imaginea se va închide).

2.4. Reglaje automate (Auto)

Meniul „Colors” conține, pe lângă uneltele de culoare discutate mai sus, și un submeniu destinat reglajelor automate. Acestea pot îmbunătăți net imaginea în multe situații, dar efectul pe

care-l produc variază mult de la o imagine la alta. In unele situații aceste reglaje pot produce o înrăutățire a imaginii procesate. Cele mai folosite reglaje automate sunt:

- *Equalize*: Acționează asupra distribuției intensității fiecărui canal în parte (prin aplicarea uneltei „*Curves*" asupra fiecărui canal) pentru ca histograma luminozității să fie cât mai plată pe tot intervalul de definire (0-255).
- *White Balance*: Extinde caracteristica fiecărui canal (prin aplicarea uneltei „*Levels*" pe fiecare canal în parte) până ce acestea ocupă în întregime intervalul lor de definire.
- *Normalize*: Acționează asupra distribuției luminozității imaginii (prin aplicarea automată a uneltei „*Levels*" pe canalul luminozității) pentru ca cel mai întunecat pixel din imagine să devină negru absolut *(#000000*) iar cel mai luminos pixel din imagine să devină alb absolut *(#FFFFFF*). Nu schimbă nuanța culorilor imaginii, deoarece normalizarea se face doar pe canalul luminozității.

3. Filtre speciale

Filtrele speciale (Figura 5) reprezintă o colecție de funcții matematice complexe ce pot fi aplicate pe imagine sau pe o regiune selectată din imagine. Aceste funcții au aplicații foarte diverse, de la începețoșarea imaginii până la re-pictarea acesteia folosind tehnici specifice unui anume curent artistic. Toate comenzile destinate aplicării acestor filtre speciale pe imagine se găsesc în meniul „*Filters*". Cele mai uzuale dintre acestea sunt enumare în Figura 5:

3.1. Reducerea și îmbunătățirea clarității

Reducerea clarității unei imagini se face folosind una din comenzile meniului „*Filters* | *Blur*". Dacă doriți doar o ușoară începețoșare a regiunii selectate, atunci cea mai rapidă cale este prin folosirea comenzii simple „*Blur*" (Figura 5). Pentru un efect mai puternic este recomandată folosirea comenzii „*Gaussian Blur*". Atât prima, cât și a doua comandă, produc o începețoșare omnidirecțională. Dacă se dorește o începețoșare după o anumită direcție trebuie folosită comanda „*Motion Blur*".

Procesul invers, de îmbunătățire a clarității unei imagini, este realizat prin una din comenzile meniului „*Filters* | *Enhance*" (Figura 5). Comenzile „*Sharpen*" și „*Unsharpen Mask*" produc efectul invers filtrului „*Blur*". Cea mai eficientă dintre ele este „*Unsharpen Mask*". Pentru a preveni schimbările de nuanță ce pot apărea în timpul aplicării acestei comenzi, este indicat ca imaginea să fie descompusă în componentele ei HSV (prin comanda „*Colors* | *Components* | *Decompose*" și selectarea modelului de culoare „HSV") iar filtrul de îmbunătățire a clarității „*Unsharpen Mask*" să fie aplicat doar asupra componentei V (*Value*). Reconstrucția imaginii se realizează ulterior prin comanda inversă „*Colors* | *Components* | *Compose*". Dacă nu dorim ca fiecare tranziție lină în imaginea inițială să fie accentuată, atunci putem mări pragul de activare („*Threshold*") din dialogul comenzii „*Unsharpen Mask*". Puterea filtrului este reglată din parametrul „*Amount*".

Filtrul „*Deinterlace*" (Figura 5) elimină defectele (liniile lipsă) în imaginile capturate după o sursă video cu baleiaj intercalat (*interlaced*). Filtrul „Despeckle" elimină zgomotul granulat care apare atunci când creștem foarte mult claritatea unei imagini sau când realizăm o fotografie digitală folosind un ISO mare. Filtrul „*Destripe*" elimină dungile verticale parazite.

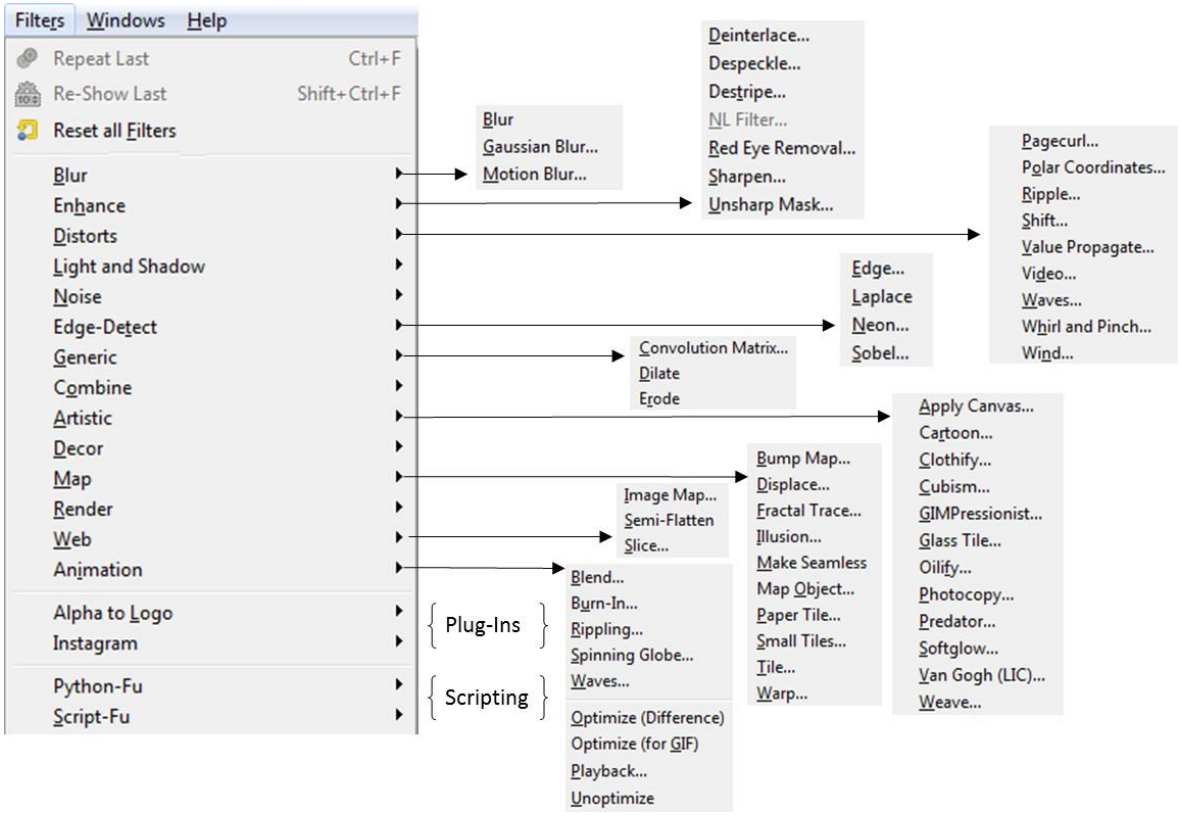


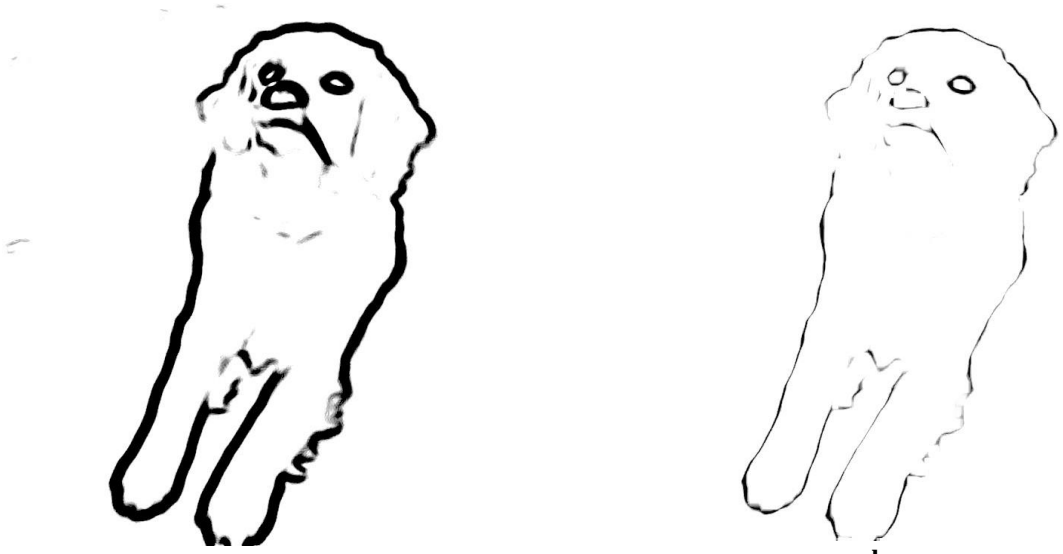
Figura 5. Meniul filtrelor speciale GIMP

3.2. Detecția marginilor și convoluția cu matrici

Pentru conturarea obiectelor dintr-o imagine putem folosi oricare dintre algoritmii de detecție a contururilor, prezenți în meniul *„Filters | Edge-Detect”*. Cei mai importanți dintre aceștia sunt enumerați în Figura 5: *Edge*, *Laplace*, *Neon*, *Sobel*. Toate aceste filtre funcționează în general prin calcularea derivatei intensității pixelilor din imagine. Filtrul *Laplace* se bazează pe calcularea derivatei de ordinul 2 și nu este în general indicat pentru detecția conturului, dar este eficient pentru mărirea contrastului.

Putem așadar să construim conturul unei imagini mult mai simplu decât am făcut-o în laboratorul trecut prin intermediul uneltelor de selecție. Să luăm exemplul din Figura 6a, unde s-a folosit filtrul *„Edge”* pentru conturarea subiectului din Figura 1a. Filtrul *„Edge”* a fost aplicat folosind parametrii standard, după conversia imaginii în nuanțe de gri (*„Colors | Desaturate”*). Dacă micșorăm parametrul *„Amount”* al filtrului *„Edge”* vom putea detecta doar schimbările abrupte de intensitate, iar conturul va fi desenat folosind linii fine. Invers, prin creșterea acestui parametru vom putea detecta toate schimbările de intensitate, iar contururile obiectelor vor fi îngroșate.

Atât funcțiile de desenare a contururilor, precum și cele de micșorare sau mărire a clarității prezentate în această secțiune, rezultă din înmulțirea matricii-imagine cu o matrice mobilă (kernel) de ponderi care se plimbă pe toată suprafața imaginii. Această operație se numește convoluție și poate fi comandată de utilizator folosind filtrul *„Filters | Generic | Convolution Matrix”*. După cum se poate vedea în Figura 7, dimensiunea maximă a matricii pe care o putem folosi drept kernel este de 5x5, dar este la latitudinea utilizatorului să aleagă care elemente ale acestei matrici să fie folosite și care nu.



a. **b.**

Figura 6. Desenarea conturului unui obiect cu GIMP: a. folosind filtrul Edge; b. după aplicarea operatorului de dilatație „Filters | Generic | Dilate”.

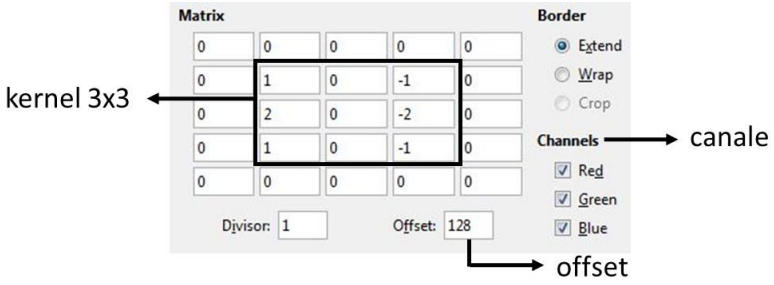


Figura 7. Folosirea matricii de convoluție pentru calcularea gradientului pe direcție orizontală.

Luminozitatea fiecărui pixel în noua imagine $(B'_{j,i})$ va fi egală cu suma luminozităților pixelilor vecini din vechea imagine $B_{j+y,i+x}$ ponderate cu elementele matricii de convoluție $(K_{y,x})$, conform formulei:

$$B'_{j,i} = \frac{\sum_{y=-2}^{2}\sum_{x=-2}^{2} K_{y,x} \cdot B_{j+y,i+x}}{Divizor} + Offset \qquad (1)$$

Dacă fixăm elementele matricii de convoluție precum în Figura 7, atunci vom realiza o derivare a imaginii pe direcție orizontală. Putem repeta operația înlocuind matricea kernel din Figura 7 cu transpusa ei, pentru a realiza derivarea pe direcție verticală. Pentru a accentua efectul derivării, putem înlocui divizorul 1 în fereastra de dialog din Figura 7 cu un divizor subunitar.

3.3. Efecte artistice și speciale

In meniul GIMP rezervat filtrelor este dedicat un număr foarte mare de comenzi efectelor artistice (Figura 5) și efectelor speciale. Comenzile submeniului „Filters | Distorts” pot genera o gamă largă de distorsiuni geometrice asupra imaginii. Cele din submeniul „Filters | Light and Shadow” sunt destinate adăugării de surse de lumină sau de umbre pe imagine.

Câteva din cele mai importante filtre ale submeniului *„Filters | Artistic”*, care este destinat realizării de efecte artistice pe imagine, sunt următoarele:

- *„Apply canvas”*: modifică textura imaginii în așa fel încât să creeze impresia că are ca suport o pânză de pictură.
- *„Cartoon”*: modifică imaginea pentru a crea impresia că face parte dintr-un desen animat.
- *„Clothify”*: modifică textura imaginii, creând impresia de țesătură
- *„Cubism”*: modifică imaginea ținând cont de tehnicile curentului artistic cubist
- *„GIMPressionist”*: este cel mai complet efect artistic. In fereastra de opțiuni a acestui efect utilizatorul poate regla textura suprafeței de pictură (tab-ul *„Paper”*), tipul de pensulă folosit (tab-ul *„Brush”*), variația unghiului de orientare al pensulei (tab-ul *„Orientation”*), mărimile pensulelor folosite (tab-ul *„Size”*), etc.

Submeniul *„Filters | Decor”* conține o colecție de scripturi (*Script-Fu*) destinate adăugării de margini și a altor câteva artefacte interesante pe imagine. Comenzile submeniului „Filters | Map” pot realiza o suprapunere a imaginii curente peste o formă predefinită sau chiar în mai multe poziții ale aceleiași imagini. Putem suprapune imaginea curentă chiar și peste o sferă, folosind comanda *„Filters | Map | Map Object”* și selectând *„Sphere”* în dreptul opțiunii *„Map to”*.

Filtrele din submeniul „Filters | Render” se diferențiază de celelalte filtre prin faptul că nu procesează conținutul aflat deja în stratul (layer-ul) selectat, ci creează noi forme din nimic, de cele mai multe ori ștergând complet conținut precedent al stratului. Este de aceea indicat ca aceste filtre (care ar trebui numite mai degrabă unelte) să fie aplicate deasupra unui strat nou.

3.4. Aplicații web

3.4.1. Harta unei imagini

Cele trei filtre ale submeniului *„Filters | Web”* sunt destinate imaginilor folosite în aplicațiile web. Primul dintre acestea, *„Image Map”* este utilizat la realizarea hărții unei imagini. Această hartă (descrisă în cod html) delimitează regiunile „calde” dintr-o imagine web. Regiunile calde sunt acele regiuni ce pot răspunde în mod independent de restul imaginii unui eveniment al mouse-ului. Un click așadar pe zona corespunzătoare unei astfel de regiuni poate deschide o nouă adresă web sau poate declanșa o funcție javascript specifică. O serie de alte schimbări pot însoți de și alte evenimente ale mouse-ului, cum ar fi modificarea cursorului mouse-ului la trecerea sa peste regiunea delimitată.

Figura 8. Crearea unei hărți de imagine pentru o pagină web folosind filtrul Filters | Web | Image Map.

Operarea cu acest filtru se realizează într-o planșă nouă de lucru. In Figura 8 putem vedea rezultatul aplicării filtrului pe imaginea din Figura 6a. In partea stângă a acestei figuri apar uneltele folosite pentru demarcarea regiunilor calde din imagine. Cea mai precisă demarcare poate fi realizată cu unealta poligon (ultima unealtă încercuită în Figura 8) . Imediat după demarcarea unei regiuni, pe ecran se va deschide o nouă fereastră responsabilă cu fixarea proprietăților noii regiuni demarcate. In dreptul opțiunii „*URL to activate...*” trebuie trecută adresa către care browserul utilizatorului va fi trimis ca urmare a unui click pe regiune. In dreptul tab-ului „JavaScript” putem fixa acțiunile browserului în cazul altor evenimente ale mouse-ului.

După demarcarea tuturor regiunilor de interes rezultatul va fi salvat într-un fișier text folosind iconița dedicată din Figura 8. Pentru a testa rezultatul se poate crea o pagină .html care să conțină doar imaginea și care să includă conținutul fișierului hartă.

3.4.2. Tăierea unei imagini

O imagine poate fi tăiată în fragmente folosind comanda „*Filters | Web | Slice*”. Fragmentele vor fi automat aranjate într-un tabel html pentru ca imaginea să poată fi reconstituită în integralitate într-o pagină web. Fiecare fragment-imagine astfel creat acționează total independent față de celelalte fragmente și poate avea proprietăți html total diferite de ale celorlalte porțiuni din imagine. De exemplu un singur fragment dintr-o imagine gif poate fi animat, sau un singur fragment poate avea transparență redusă, sau putem încercui cu un cadru fragmentul peste care are lor trecerea mouse-ului, etc. Inainte însă de a rula filtrul „*Slice*” este nevoie să divizăm imaginea folosindu-ne de comenzile disponibile în submeniul „*Image | Guides*”. Să luăm exemplul prezentat în Figura 9. Intâi s-a realizat selectarea capului câinelui folosind unealta de selecție rectangulară. Ulterior s-au trasat ghidurile folosind comanda „*Image | Guides | New Guides from Selection*”. Ghidurile realizează demarcarea fragmentelor.

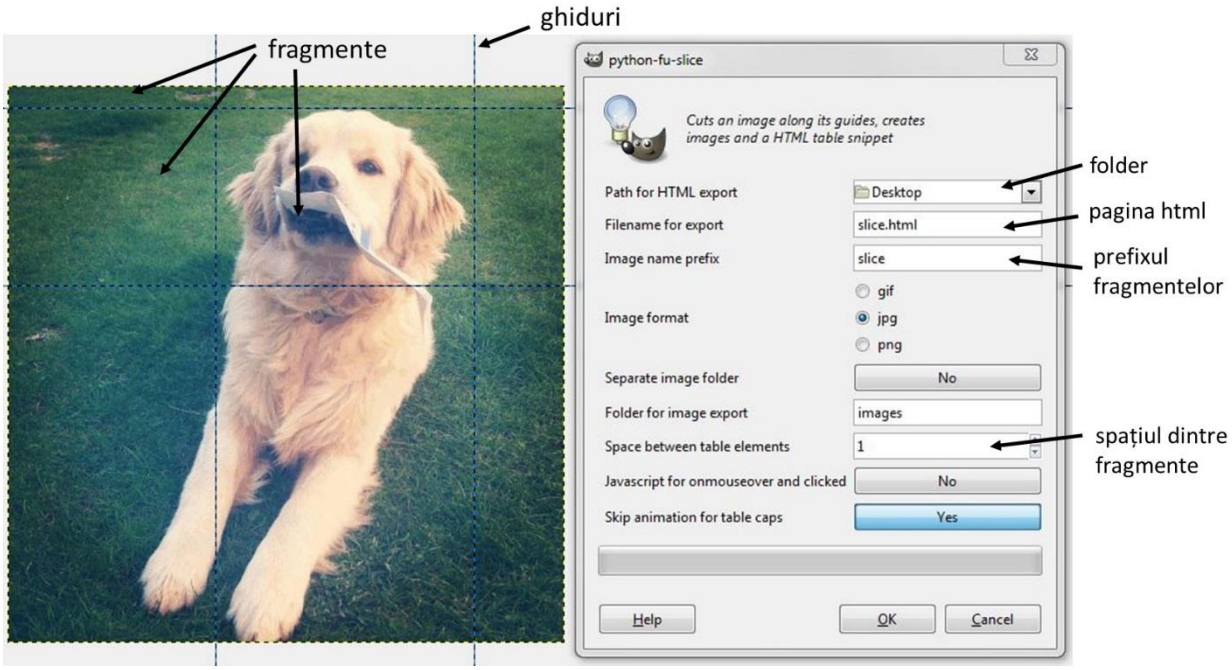


Figura 9. Tăierea unei imagini în fragmente pentru web folosind filtrul „Filters | Web | Slice”.

Pentru a face fragmentarea propriu-zisă a imaginii, putem porni acum filtrul „*Image | Guides | Slice*”. In fereastra de dialog care se va deschide trebuie introdus numele folderului unde întreaga structură a noii pagini web va fi copiată, urmat de numele paginii web. Pentru cele 9 fragmente ale imaginii din Figura 9 trebuie apoi introdus un identificator text comun, care se va constitui în prefixul numelui tuturor celor 9 imagini independente create. Se mai poate fixa și spațiul dintre celule tabelului ce vor conține fragmentele din imagine. Dacă este selectată opțiunea „*Skip animation for table caps*” atunci fragmentele situate la frontiera imaginii nu vor răspunde la evenimentele mouse-ului asupra lor.

4. Script-Fu

Script-Fu este un limbaj de scripting derivat din limbajul Scheme care poate fi folosit pentru generarea automată a unui număr mare de operații grafice. Este în special util când se dorește repetarea identică a unei suite de operații pe mai multe imagini. Prin folosirea unui script în locul repetării manuale a operațiilor putem câștiga timp și putem de asemenea împărtăși foarte rapid tehnică grafică altor utilizatori. Se pot de asemenea downloada de pe internet filtre grafice create de alți utilizatori GIMP, fie sub formă de Script-Fu, fie sub formă de Plug-Ins (scrise în general într-un limbaj de programare precum C, folosind librării specifice). In cazul în care ați downloadat un script, acesta trebuie copiat în folderul dedicat, care poate fi găsit folosind comanda „*Edit | Preferences*”, urmată de selecția opțiunii „*Folders | Scripts*”. In cazul unui Plugin, acesta trebuie copiat în folderul selectat de opțiunea „*Folders | Plug-Ins*”. Pentru acutalizarea listei de scripturi este nevoie de rularea comenzii „*Filters | Script-Fu | Refresh Scripts*”.

Există două categorii de scripturi: cele care creează o nouă imagine (pe care în general le putem găsi în submeniul „*File* | *Create*”) și cele care operează pe imaginea deja încărcată (pe care le putem găsi în interiorul submeniului „*Filters*”). Ambele categorii de scripturi pot fi găsite pe disc la aceeași locație (fixată de opțiunea „*Folders* | *Scripts*” discutată în paragraful precedent). Dacă deschidem locația respectivă de pe disc vom găsi lista tuturor scripturilor instalate. Avem posibilitatea să vedem codul oricăruia dintre ele folosid un editor de text (*Notepad++*). Putem așadar copia codul oricărui script deja existent și crea propriul nostru script plecând de la acesta.

4.1. Scrierea unui script

Pentru a testa comenzile Script-Fu primul pas pe care trebuie să-l facem este să pornim consola limbajului Scheme. Aceasta se realizează folosind comanda „*Filters* | *Script-Fu* | *Console*”. Fiecare comandă sau operație matematică va fi scrisă între paranteze rotunde *( )*. Primul termen după deschiderea parantezei va fi numele comenzii sau operatorul matematic folosit, urmat de un spațiu alb (obligatoriu). Imediat urmează seria operanzilor, separați de spații albe. Pentru scrierea operațiilor matematice de exemplu, comenzile vor arăta precum în Figura 10. Se poate observa că fiecare operator matematic poate accepta mai mult de 2 operanzi și de asemenea că fiecare operand poate la rândul lui să fie rezultat în urma unei operații matematice separate (scrisă între două paranteze interioare). Rezultatul fiecărei operații apare scris pe linia imediat următoare, după apăsarea tastei „*Enter*” (Figura 10).

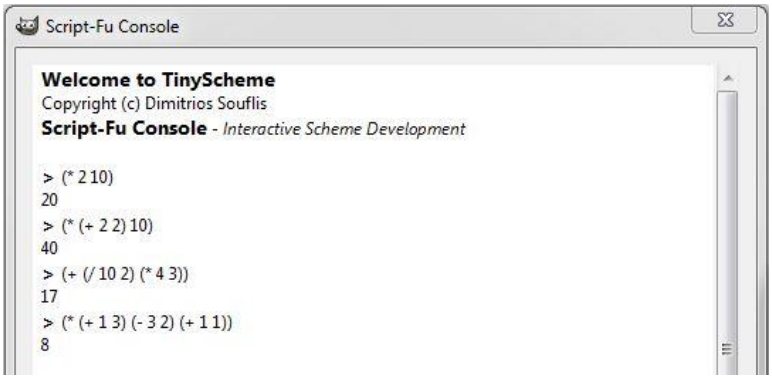


Figura 10. Folosirea operatorilor matematici în consola Script-Fu

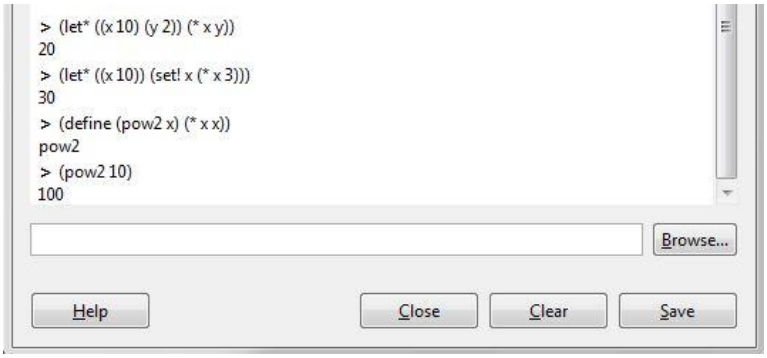


Figura 11. Definirea variabilelor și a funcțiilor în consola Script-Fu

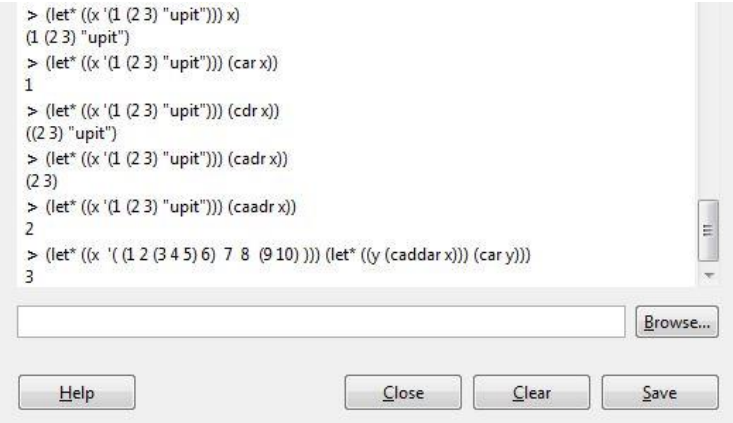


Figura 12. Folosirea listelor în Script-Fu

Pentru definirea unei variabile se folosește comanda *let** a cărei utilizare este exemplificată în Figura 11, folosind consola. Doar atunci când folosim consola este nevoie să scriem toată paranteza pe un rând. Când vom realiza un script într-un fișier Script-Fu având terminația *.scm,* atunci putem desfășura conținutul unei paranteze pe mai multe rânduri. Pe prima linie din Figura 11 am inițializat așadar variabila x la valoarea 10 și variabila y la valoarea 2, folosind comanda *let**. In aceeași paranteză care cuprinde comanda *let**, am scris și operatația de înmulțire *(* x y)*. Aceasta pentru că variabilele x și y nu au sens în afara parantezei care conține comanda *let**. Pentru a schimba valoarea unei variabile după ce aceasta a fost definită, putem folosi comanda *set!*, precum în a doua linie de comandă din Figura 11.

Pentru definirea unei funcții se folosește comanda „*define*”, urmată de o paranteză care va conține numele funcției și numele tuturor parametrilor funcției. Uitându-ne la a treia linie din Figura 11, noua funcție poartă numele de „*pow2*” și primește un singur parametru „*x*”. Ultima operație înainte de închiderea parantezei funcției fixează valoarea returnată. Pentru cazul funcției *pow2,* aceasta va returna valoarea *(* x x)*, adică x^2. Funcția poate fi testată apoi într-o comandă separată, precum în ultima linie de comandă din Figura 11.

Putem defini o variabilă (de exemplu *x*) nu doar sub forma unei valori unitare, ci și ca un vector (listă). Lista este o structură de date tipică limbajului Scheme și poate fi creată precum în prima linie din Figura 12. Se poate observa cum variabilei *x* i se alocă o listă definită ca o paranteză precedată de un apostrof. Observăm că al doilea element al listei este o altă listă. De data aceasta, fiind vorba de o listă interioară, nu este necesară folosirea apostrofului. Mai putem observa că listele pot conține variabile de tipuri diferite. Intr-adevăr, spre deosebire de limbaje precum C sau Java, în Scheme nu este nevoie să fim preocupați de tipurile de date. Pentru a accede la primul element al listei, s-a folosit în a doua linie comanda *„car”*. Pentru a accesa restul listei se folosește comanda *„cdr”* (a treia linie din Figura 12). Există comenzi care înlănțuiesc cele două instrucțiuni, cum este exemplificat în ultimele linii de comandă din Figura 12. De exemplu, o comandă „*car*” aplicată valorii returnate de o comandă „*cdr*” se scrie prescurtat drept „*cadr*” precum în a patra linie de comandă din Figura 12.

4.2. Un script de animație

Putem genera o animație gif folosind un filtru creat de noi, asemănător celei pe care am realizat-o folosind filtrul nativ „*Filters* | *Animation* | *Waves*” (exercițiul 2.6.7). Vom face aceasta

analizând codul filtrului nativ, schimbându-l acolo unde este necesar. Primul pas este așadar copierea filtrului „*waves-anim.scm”* într-un fișier *„whirl-anim.scm”*. Dacă studiem codul filtrului inițial putem observa că la un moment dat în cod se face apel la un plug-in numit „*plug-in-waves*”. Acest plug-in poate fi apelat și direct de utilizator folosind comanda *„Filter | Distorts | Waves”*. Filtrul de animație *„waves-anim.scm”* instanțiază așadar imaginea într-o serie de cadre identice, iar apoi va aplica fiecărui cadru în parte plugin-ul „*Waves*” folosind de fiecare dată alți parametri. Atunci când salvăm noua serie de cadre în format gif, schimbările produse fiecărui cadru se vor traduce în iluzia propagării unor valuri pe suprafața apei.

Noul filtru („*whirl-anim.scm”*) va face apel la o altă comandă, anume *„Filter | Distorts | Whirl and Pinch”*, pentru a genera efectul de animație. Este bine ca înainte de a începe să programăm noul filtru de animație să ne familiarizăm cu parametrii pe care-i poate primi această comandă (*„Whirl and Pinch”)* și să-i comparăm cu parametrii pe care-i primește plugin-ul *„Waves”*. Observăm așadar că plugin-ul „*Filter | Distorts | Whirl and Pinch*” poate primi doar trei parametri: *Whirl angle, Pinch amount* și *Radius*. Filtrul de animație pe care-l vom scrie în această secțiune va varia doar primul dintre cei trei parametri (*Whirl angle*). Ceilalți doi parametri se vor păstra constanți în toate cadrele generate de filtrul de animație.

Primele modificări pe care trebuie să le aducem filtrului „waves-anim” pentru a-l adapta scopului nostru sunt prezentate în Figura 13. Ultima funcție (*script-fu-menu-register*) fixează plasamentul în meniu unde vom putea găsi filtrul nou creat. Funcția *script-fu-register* fixează ce parametri vor putea fi reglați în dialogul filtrului (fereastra care se deschide imediat după selectarea filtrului și care precede execuția sa).

```
(script-fu-register "script-fu-waves-anim"
  _"_Waves..."
  _"Create a multi-layer image with an effect like a stone was
  "Sven Neumann <sven@gimp.org>"
  "Sven Neumann"
  "1997/13/12"
  "RGB* GRAY*"
  SF-IMAGE       "Image" 0
  SF-DRAWABLE    "Drawable" 0
  SF-ADJUSTMENT _"Amplitude"        '(10 1   101 1 10 1 0)
  SF-ADJUSTMENT _"Wavelength"       '(10 0.1 100 1 10 1 0)
  SF-ADJUSTMENT _"Number of frames" '(6  1   512 1 10 0 1)
  SF-TOGGLE     _"Invert direction" FALSE
)

(script-fu-menu-register "script-fu-waves-anim"
                         "<Image>/Filters/Animation/Animators")
```

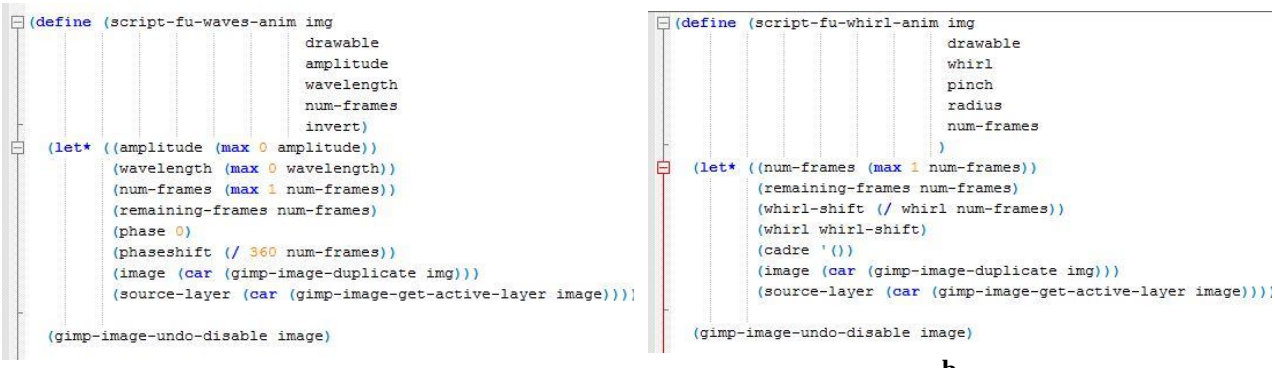
a.

```
(script-fu-register "script-fu-whirl-anim"
  "whirl"
  "Creates a multi-layer image with a whirling effect"
  "Adrian Iordachescu <adi_iord@yahoo.com>"
  "Adrian Iordachescu"
  "2014/12/01"
  "RGB* GRAY*"
  SF-IMAGE       "Image" 0
  SF-DRAWABLE    "Drawable" 0
  SF-ADJUSTMENT "Whirl"        '(10 0 720 1 10 2 0)
  SF-ADJUSTMENT "Pinch"       '(0 -1 1 0.1 1 3 0)
  SF-ADJUSTMENT "Radius" '(1 0 2 0.1 1 3 0)
  SF-ADJUSTMENT "Number of frames" '(6  1   512 1 10 0 1)
)

(script-fu-menu-register "script-fu-whirl-anim"
                         "<Image>/Filters/Animation/Adrian")
```

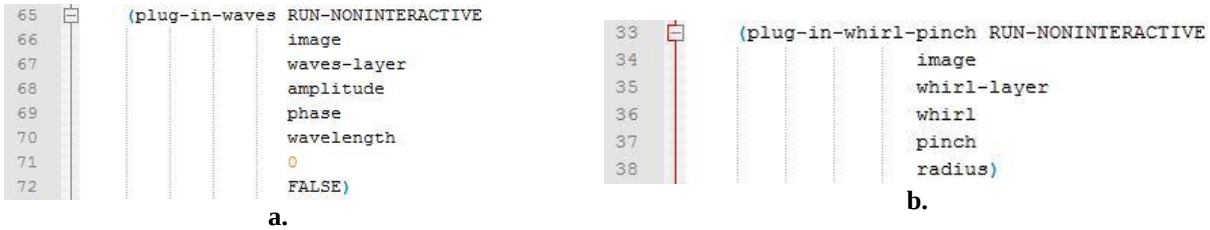
b.

Figura 13. Pasul 1: Transformarea filtrului nativ „waves-anim.scm” (a) într-un nou filtru de animație „whirl-anim.scm” (b)



a. **b.**

Figura 14. Pasul 2: Transformarea filtrului nativ „waves-anim.scm”(a) într-un nou filtru de animație „whirl-anim.scm” (b)



a. **b.**

Figura 15. Pasul 3: Transformarea filtrului nativ „waves-anim.scm”(a) într-un nou filtru de animație „whirl-anim.scm” (b)

Parametrii proprii noului filtru (*Whirl, Pinch* și *Radius*) sunt toți trei reglați folosind un control de tipul „*SF-ADJUSTMENT*” (Figura 13b). Acest tip de control primește întotdeauna ca prim parametru șirul de caractere care va fi afișat în fereastra de dialog în dreptul său, urmat de o listă ce va conține valoarea inițială a parametrului, valoarea minimă, valoarea maximă, pasul, pasul mărit (prin apăsarea butoanelor pageUp/pageDown), numărul de zecimale după virgulă și modul de lucru al controlului (0 sau 1). Numărul de cadre al animației noastre este reglat folosind tot un control de tipul „*SF-ADJUSTMENT*”, lucrând în modul de lucru restrâns (ultimul parametru este 1, în loc de 0).

După modelarea dialogului de inițializare, urmează apelarea funcției propriu-zise de animație. Observăm în Figura 14 că parametrii funcției sunt identici cu parametrii folosiți la modelarea dialogului de inițializare. Urmează definirea unor variabile locale, urmărind regulile de sintaxă enunțate în secțiunea 3.1.

Urmează modificarea comenzii plug-in ce este executată pe fiecare cadru al animației în parte (Figura 15). Numele plugin-ului poate fi citit din folderul de plugin-uri. Numele plug-inului dedicat comenzii „*Waves*” este „*waves.exe*”, iar cel al plugin-ului dedicat comenzii „*Whirl and Pinch*” este „*whirl-pinch.exe*”. Primii trei parametri ai plugin-urilor sunt identici: modul de lucru, imaginea curentă și cadrul activ. Următorii parametri sunt specifici fiecărui filtru în parte.

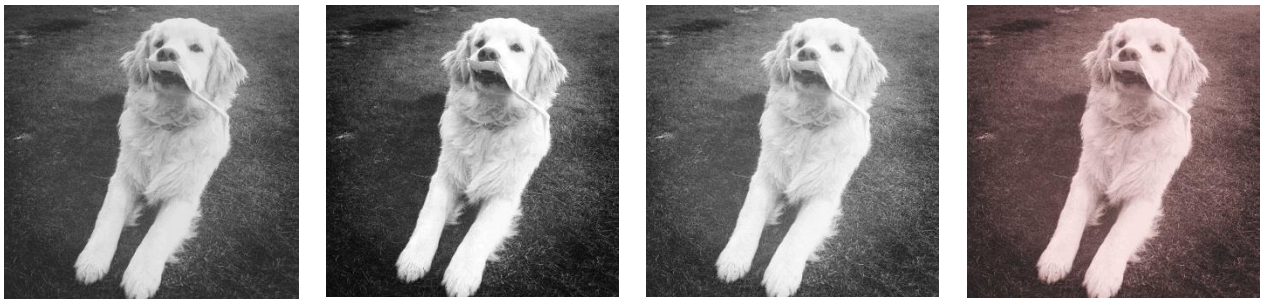
Codul explicat al noului filtru de animație „*Whirl*” este afișat în întregime în anexa acestei lucrări de laborator. Dacă faceți o comparație între acesta și codul filtrului nativ „*Waves*” veți putea observa că noul filtru va instanția un număr dublu de cadre decât o face filtrul nativ. Numărul suplimentar de cadre este necesar pentru a asigura o tranziție fină între cadrul în care parametrul de răsucire (*whirl*) este maxim și cadrul inițial (în care răsucirea este nulă). Acest număr suplimentar de cadre nu era necesar pentru cazul filtrului „*Waves*”, al cărui parametru variabil era faza (parametru periodic).

5. Desfășurarea lucrării

5.1. Incercați să repetați cele două imagini obținute în Figura $1_{b,c}$, folosindu-vă atât de unealta „Color Balance" cât și de unealta „Hue-Saturation". Inregistrați rezultatele obținute în patru fișiere .jpg distincte.

5.2. Transformați în alb-negru imaginea din Figura 1a folosind comanda „Colors | Desaturate" (Figura 16a). Folosiți apoi unealta „Levels" pentru a transforma liniar spațiul luminozităților de intrare din intervalul 28-238 în intervalul de ieșire 0-255 (Figura 16b). Ce modificări apar pe histograme după această transformare? Pentru a accesa histogramele culorilor și intensității luminoase, este suficientă folosirea comenzii din meniu „*Colors* | *Info* | *Histogram*", utilizată și în prima lucrare de laborator. Repetați din nou aceeași transformare însă folosind de data aceasta pentru parametrul „*gamma*" valoarea de 1,5 (Figura 16c). Studiați din nou histograma fiecărui canal în parte și observați diferențele. Reveniți la imaginea din Figura 16a. Aplicați din nou transformarea de la punctul precedent ((Figura 16c), însă selectând în locul canalului luminozităților (*Value*), canalul nivelurilor de roșu (*Red*). Imaginea care se va obține este cea din Figura 16d. Studiați din nou histogramele tuturor canalelor noii imagini.

5.3. Faceți o comparație între histogramele culorilor și intensității luminoase înainte și după aplicarea fiecărui reglaj automat discutat în Secțiunea 1.5 pe imaginea din Figura 1a.



a. b. c. d.

Figura 16. Folosirea uneltei "Levels" pentru editarea imaginii din Figura 1a: a. imaginea originală desaturată; b. după aplicarea uneltei Levels pe imaginea de la punctul „a" folosind o transformare liniară a intervalului 28-238 în intervalul 0-255; c. după aplicarea aceleiași transofmări de la punctul „b", însă folosind un gamma de 1,5; d. după aplicarea aceleiași transformări de la punctul „c", însă pe canalul roșu

5.4. Folosiți-vă de uneltele de culoare *Levels („Colors | Levels")*, *Desaturate („Colors | Desaturate")* și *Invert („Colors | Invert")* precum și de filtrele Blur („Filters | Blur | Gaussian Blur") și *Edge („Filters | Edge-Detect | Edge")* pentru a transforma imaginea din Figura 1a în imaginea din Figura 6a.

5.5. Folosiți-vă de filtrul de dilatație *(„Filters | Generic | Dilate")* și de uneltele de selecție pentru a transforma imaginea obținută la exercițiul precedent (Figura 6a) în imaginea din Figura 6b.

5.6. Folosiți matricea de convoluție din Figura 7 pentru a obține harta gradientului (atât pe orizontală cât și pe verticală) a imaginii din Figura 1a. Ce se întâmplă dacă înjumătățim divizorul în fereastra de dialog din Figura 7?

5.7. Creați un puzzle din 49 de bucăți pe baza imaginii din Figura 1a folosind comanda *„Filters | Render | Pattern | Jigsaw”*.

5.8. Realizați o hartă de imagine pentru poza din Figura 6a. Selectați 4-5 regiuni de interes din image și instruiți browserul ca în urma unui click pe una dintre aceste regiuni să răspundă cu o fereastră de dialog (*javascript: alert('text')*) în care să fie afișată denumirea regiunii respective. Rezultatul va fi integrat într-o pagină web și testat în browser.

5.9. Realizați meniul unei pagini web personalizate sub forma unei imagini originale pe care să o fragmentați în mai multe zone independente corespunzătoare etichetelor: *Home*, *History*, *Contact*, etc., care să-și schimbe transparența la trecerea mouse-ului peste ele.

5.10. Folosind instrucțiuni de tip *car*, *cdr* și înlănțuirile lor, accesați elementul „6” din ultima listă inițializată în Figura 12: *((1 2 (3 4 5) 6) 7 8 (9 10))*.

5.11. Implementați scriptul de animație „*whirl-anim.scm*” explicat în această secțiune și aplicați-l pe o imagine aleasă de dumneavoastră pentru generarea unui fișier *gif* de cel puțin 20 de cadre. Animația trebuie să pornească de la un unghi minim de 0°, să ajungă la un unghi de răsucire de 600° și să se întoarcă treptat la unghiul inițial de 0° înainte de a se repeta.

5.12. Notați dimensiunea fișierului *gif* obținut la exercițiul precedent. Aplicați acum pe imaginea *gif* obținută anterior filtrul *„Filters | Animation | Optimize (for GIF)”* și salvați din nou imaginea într-un nou fișier *gif*. Cum s-a modificat dimensiunea noii imagini? Cum vă explicați modificarea?

5.13. Realizați un filtru de animație bazat în continuare pe plugin-ul „*Whirl and Pinch”*, de data aceasta însă păstrând răsucirea constantă și variind doar strangularea (parametrul *pinch*). Aplicați filtrul astfel implementat pe o imagine aleasă de dumneavoastră. Animația trebuie să pornească de la o strangulare nulă și să ajungă la o strangulare maximă de 0,5 (*pinch=0.5*). Revenirea la cadrul inițial trebuie să se facă treptat. Animația astfel creată se va repeta în buclă.

ANEXA 1 – CODUL INTEGRAL AL FILTRULUI DE ANIMATIE „WHIRL”

```
(define (script-fu-whirl-anim img
                              drawable
                              whirl
                              pinch
                              radius
                              num-frames
                             )
  (let* ((num-frames (max 1 num-frames))
         (remaining-frames num-frames)
         (whirl-shift (/ whirl num-frames))
         (whirl whirl-shift)
         (cadre '())
         (image (car (gimp-image-duplicate img)))
         (source-layer (car (gimp-image-get-active-layer image))))

    (gimp-image-undo-disable image)
```

```
(while (> remaining-frames 1) ; cat timp mai raman cadre de prelucrat
  (let* (
        (whirl-layer (car (gimp-layer-copy source-layer TRUE))) ; copiaza cadrul sursa intr-un nou cadru
        (layer-name (string-append "Frame "   ; denumirea cadrului nou este inregistrata intr-un sir de caractere
                                   (number->string ; introducerea in numele cadrului a unui numar de ordine
                                     (- (+ num-frames 2)
                                        remaining-frames) 10
                                     )
                                   " (first)")) ; introducerea sirului first semnifica sensul rasucirii: crestere
        )
  (gimp-layer-set-lock-alpha whirl-layer FALSE) ;
  (gimp-image-insert-layer image whirl-layer 0 -1) ; urcarea noului cadru in stiva de cadre a imaginii pe prima pozitie (-1)
  (gimp-item-set-name whirl-layer layer-name) ; botezarea noului cadru folosind sirul de caractere layer-name

  (plug-in-whirl-pinch RUN-NONINTERACTIVE ; apelarea plugin-ului whirl-pinch
                image  ; parametru general: imaginea
                whirl-layer ; al doilea parametru general: stratul
                whirl ; parametru special: unghiul de rasucire
                pinch ; parametru special: strangularea
                radius) ; parametru special: raza efectului

  (set! remaining-frames (- remaining-frames 1)) ; scaderea numarului de cadre ramase pentru a fi procesate
  (set! whirl (+ whirl whirl-shift)) ; modificarea unghiului de rasucire
  (set! cadre (cons (car (gimp-layer-copy whirl-layer TRUE)) cadre)) ; adaugarea cadrului prezent intr-o stiva virtuala de cadre
  )
 )

 (set! remaining-frames (- num-frames 1)) ; reinitializarea numarului de cadre ramase pentru a fi procesate

  (while (> remaining-frames 1) ; cat timp mai raman cadre de prelucrat
  (set! cadre (cdr cadre)) ; scoaterea unui cadru din stiva virtuala de cadre
  (let* (
        (whirl-layer (car cadre)) ; citirea primului cadru din stiva

        (layer-name (string-append "Frame " ; denumirea noului cadru va fi inregistrata intr-un sir de caractere
                                   (number->string
                                     remaining-frames
                                     10
                                     )
                                   " (second)")) ; introducerea sirului second semnifica sensul rasucirii: descrestere
        )
  (gimp-layer-set-lock-alpha whirl-layer FALSE) ;
  (gimp-item-set-name whirl-layer layer-name) ; botezarea noului cadru folosind sirul de caractere layer-name
  (gimp-image-insert-layer image whirl-layer 0 -1) ; urcarea noului cadru in stiva de cadre a imaginii pe prima pozitie (-1)

  (set! remaining-frames (- remaining-frames 1)) ; scaderea numarului de cadre ramase pentru a fi procesate
  )
 )

 (gimp-item-set-name source-layer "Frame 1") ; botezarea cadrului sursa folosind sirul de caractere Frame 1
 (gimp-image-undo-enable image)
 (gimp-display-new image) ; afiseaza noua imagine

 )
)

(script-fu-register "script-fu-whirl-anim"
  "whirl"
  "Creates a multi-layer image with a whirling effect"
  "Adrian Iordachescu <adi_iord@yahoo.com>"
  "Adrian Iordachescu"
  "2014/12/01"
  "RGB* GRAY*"
  SF-IMAGE       "Image" 0
  SF-DRAWABLE    "Drawable" 0
  SF-ADJUSTMENT "Whirl"        '(10 0 720 1 10 2 0)
  SF-ADJUSTMENT "Pinch"       '(0 -1 1 0.1 1 3 0)
  SF-ADJUSTMENT "Radius" '(1 0 2 0.1 1 3 0)
  SF-ADJUSTMENT "Number of frames" '(6  1   512 1 10 0 1)
)

(script-fu-menu-register "script-fu-whirl-anim"
                         "<Image>/Filters/Animation/Adrian")
```