



POLITECHNIKA WARSZAWSKA
WYDZIAŁ ELEKTRYCZNY



PRACA DYPLOMOWA MAGISTERSKA

Poprawa jakości obrazów cyfrowych poprzez metodę wmalowywania

Autor:

Marek Szweda

Promotor: dr Sławomir Skoneczny

Warszawa, listopad 2018

.....

podpis promotora

.....

podpis autora

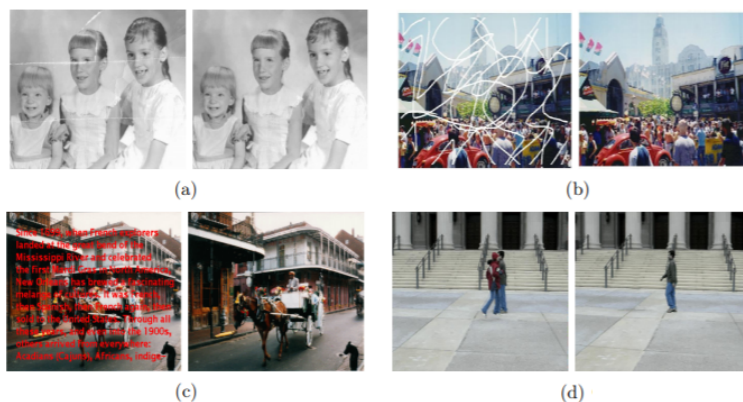
Streszczenie

Tu będzie jakieś streszczenie

Rozdział 1

Wprowadzenie

Poprawa jakości obrazów cyfrowych poprzez metodę wmalowywania w ostatnich czasach stała się jednym z głównych problemów w przetwarzaniu obrazów oraz widzeniu maszynowym. Jest szeroko stosowana w aplikacjach do edycji obrazów, usuwania niechcianych obiektów z obrazów, tekstów, zarysowań, uszkodzeń, poprawy starych zdjęć oraz filmów. Wmalowywanie obrazów definiuje się jako automatyczne wypełnianie braków w obrazie na podstawie otoczenia, w sposób uniemożliwiający obserwatorowi znalezienie pierwotnie wybrakowanych miejsc. Przykłady obrazów poddanych operacji wmalowywania przedstawia Rysunek 1.1



Rysunek 1.1: Przykłady obrazów poddanych operacji wmalowywania.

Obrazy przedstawiono w parze: po lewej obraz wejściowy, po prawej wynikowy obraz poddany operacji wmalowywania. Mogą nimi być uszkodzenia jak w przy-

padku a oraz b, napisy znajdujące się na obrazie - c, czy też obiekty, którym w konkretnym przypadku d jest postać. Przestrzeń do wmalowywania w obrazie wyznaczana jest manualnie poprzez użytkownika definiującego konkretny obiekt, który powinien według niego zostać usunięty. Program automatycznie nie jest w stanie wyznaczyć części, która powinna zostać przearanżowana.

Techniki wmalowywania znajdują szeroki zakres zastosowań w przetwarzaniu nagrań filmowych, gdzie nieznany region na scenie musi zostać oszacowany w pewien sposób dostarczając spójności wizualnej użytkownikowi. Przykładami mogą być konieczność ukrywania błędów nagrań, strat wynikających z błędów urządzeń nagrywających, czy też usunięcia niepożądanych obiektów, które w trakcie nagrywania pojawiły się w kadrze.

Głównym celem pracy jest osiągnięcie algorytmu skutecznie odbudowującego uprzednio zdefiniowaną nieznaną przestrzeń obrazu. W tym celu dokonano przeglądu istniejących algorytmów wmalowywania opartych na rozwiązaniu odpowiednio zdefiniowanych problemów matematycznych. Wyniki badanych zagadnień używane są na podstawie danych wejściowych w postaci zdefiniowanych obszarów obrazu otaczających brakujące regiony. Podstawowymi operacjami wykorzystywanymi w bardziej złożonych algorytmach rekonstrukcji są:

1. Rozwiązanie cząstkowych równań różniczkowych wyższego rzędu bazujących na kontynuacji tekstur, geometrii, struktur obrazu.
2. Problem minimalizacji energii o bogatym asortymencie sposobu definicji.
3. Wyznaczenie podobieństwa pomiędzy obszarami w obrazie za pomocą wprowadzonej metryki
4. Obliczenia przeprowadzane na modelu wariacyjnym

W pracy opisano przedstawione powyżej algorytmy, porównano oraz zaproponowano nowy model uzupełniania brakujących danych. Porównania dokonano na podstawie dwóch najistotniejszych właściwości: szybkości algorytmu oraz obiektywnej oceny poprawności otrzymanego wyniku. Zwrócono także uwagę na różne

cechy obrazu, które mogą determinować dobór algorytmu do zadanego problemu. Warto zaznaczyć, iż do tej pory nie została przedstawiona jednoznaczna definicja problemu matematycznego pozwalająca w niezawodny sposób wyznaczyć braki w obrazie niezależnie od jego cech charakterystycznych.

Rozdział 2

Przegląd istniejących algorytmów.

Pierwszą gałęzią algorytmów zgodnie z rozróżnieniem wprowadzonym w [1] są metody opierające się na rozwiązaniu równań różniczkowych wyższego rzędu. Wśród nich pojawia się próba przystosowania procesu dyfuzji do wypełniania braków w obrazie (źródła [2] lub [3]), dzięki której przy wykorzystaniu zależności z działu mechaniki płynu można dokonać próby kontynuacji informacji z otoczenia w brakujące obszary. Innym sposobem wyznaczenia braków opierającym się na równaniach różniczkowych wyższego rzędu jest próba wprowadzenia problemu minimalizacji energii zdefiniowanej w oparciu o pojęcie wahanja funkcji tak jak [4] czy też [5].

Drugą gałęzią algorytmów są metody oparte na uzupełnianiu braków obrazu kawałek po kawałku skopiowanymi częściami istniejącego obrazu z miejsc o najlepszym dopasowaniu w miejsce braku. Pierwszy raz metoda syntezy obrazu kopiowanymi kawałkami została zaproponowana w [6]. Kolejność wypełniania braków nie jest obojętna i ma znaczny wpływ na ostateczną spójność obrazu. Jedną z najpopularniejszych metod określania priorytetu została przedstawiona w [7]. Na przestrzeni czasu pojawiło się wiele metod minimalizujących zjawisko niespójności w obrazie takich jak np. [8].

Ostatnią grupą algorytmów jest grupa stanowiąca połączenie powyższych algorytmów. Najprostszym przykładem grupy może być [9], w którym zaproponowano poprzedzenie procesu wmalowywania segmentacją obrazu. Uzyskaną w procesie segmentacji teksturę uzupełnia się metodą najbliższego dopasowania nato-

miast strukturę uzyskuje na podstawie równań różniczkowych. Kolejnym przykładem może być publikacja [10], w której autorzy proponują nielokalny model wyznaczający nieznany obszar obrazu wykorzystując rachunek wariacyjny, połączony ze zdefiniowaną metryką wyznaczającą odległości pomiędzy rozpatrywanymi skrawkami obrazu.

Większość metod wmalowywania opisanych w literaturze opiera się na algorytmach kontynuujących strukturę, geometrię bądź teksturę obrazu. Modele bazujące na geometrii i strukturze w uogólnieniu generują efekt uboczny w postaci rozmytego wynikowego obrazu, bez zachowanych tekstur. Do algorytmów broniących się przed efektem rozmycia wyniku można zaliczyć takie, w których wyznaczeniu konkretnego miejsca w nieznanym obszarze może towarzyszyć przebadanie całego zbioru punktów niemodyfikowanej części obrazu. Stąd metody kontynuujące tekstury w większości uważa się za metody nielocalne.

Rozdział 3

Równania różniczkowe wyższego rzędu.

3.1 Rozwiązanie równania Naviera-Stokesa

W celu dokładnego nakreślenia idei wmalowywania w oparciu o równanie Naviera-Stokesa należy rozważyć poniższe równanie przepływu cieczy Newtonowskiej:

$$V\nabla V + \frac{dV}{dt} = -\frac{1}{q}\nabla p + F + \nu\nabla^2 V \quad (3.1)$$

$$V = \left(-\frac{\partial \Psi}{\partial y}, \frac{\partial \Psi}{\partial x}\right) \quad (3.2)$$

gdzie V oznacza prędkość lokalną, t chwilę czasu, q gęstość lokalną, p ciśnienie lokalne, F siłę zewnętrzną działającą lokalnie na ciecz, ν lepkość kinematyczną lokalną, a Ψ linię prądu. Człon $V\nabla V$ odpowiada konwekcji lokalnego pędu wraz z ruchem cieczy, $\frac{1}{q}\nabla p$ wpływowi sił ciśnieniowych na zmianę pędu (kompensuje niezerową konwekcję), F siłę zewnętrzną działającą na ciecz (np. grawitację) oraz $\nu\nabla^2 V$ dysypacji pędu cieczy od tarcia pomiędzy cząstkami. Traktując płyn jako nieściśliwy ($q \rightarrow \infty$) oraz pomijając działanie sił zewnętrznych ($F = 0$) równanie można uprościć do postaci:

$$V\nabla V + \frac{dV}{dt} = \nu\nabla^2 V \quad (3.3)$$

Zgodnie z [11] dokonując obustronnej rotacji równania otrzymujemy równanie w zależności od wirowości:

$$V\nabla\omega + \frac{d\omega}{dt} = \nu\nabla^2 V \quad (3.4)$$

gdzie wirowość określa się wzorem:

$$\omega = \nabla \times V \quad (3.5)$$

Zgodnie z [12] niech I reprezentuje obraz cyfrowy będący macierzą o rozmiarze $m \times n$. Wtedy $I_{i,j}$ zawiera wartość dodatnią całkowitą z przedziału od 0 do 255, stanowiącą informację o poziomie jasności danego piksela z obrazu. Wartość 0 opowiada odcieniu czarnemu, natomiast 255 odcieniu czystemu białemu. Niech D stanowi zbiór indeksów $(i, j) \in \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$. Obraz I wtedy można traktować jako funkcję $I : D \rightarrow R$.

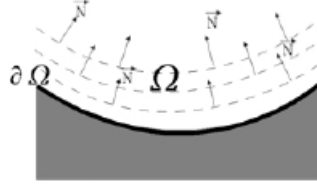
W 2000 roku Bertalmio w [3] wprowadza analogię pomiędzy wirowością a smukłością obrazu. Dokładniej przedstawia to Tabela 3.1: gdzie $\nabla^\perp = (\frac{\partial}{\partial y}, -\frac{\partial}{\partial x})$. Niech

Navier-Stokes	Wmalowywanie w obrazie
Linia prądu Ψ	Jasność obrazu I
Prędkość płynu $V = \nabla^\perp \Psi = [u, v]$	Prędkość płynu $V = \nabla^\perp I = [u, v]$
Wirowość $\omega = \nabla^2 \Psi$	Smukłość $\omega = \nabla^2 I$
Lepkość płynu ν	Parametr dyfuzji anizotropowej ν

Tabela 3.1: Mapowanie zmiennych w zagadnieniu wmalowywania obrazu.

$\Omega \subset D$ stanowi wypełniany obszar obrazu (maskę). W wyniku rozwiązania dyskretnego równania Naviera-Stokesa w przestrzeni maski z warunkami brzegowymi zdefiniowanymi jako jasność pikseli sąsiadujących z obszarem maski uzyskuje się kontynuację linii o podobnej jasności (izolinii) w głąb odrestaurowywanego obszaru zgodnie z Rysunek 3.1. Matematycznie kierunek izolinii jasności obrazu wyznacza $\nabla^\perp I$. W oparciu o Tabela 3.1 oraz (3.4) zagadnienia wmalowywania można sformułować jako problem stabilnego rozwiązania poniższego równania różniczkowego:

$$(\nabla^\perp I) \cdot \nabla (\nabla^2 I) + \frac{d(\nabla^2 I)}{dt} = \nu \nabla (g(\nabla(\nabla^2 I)) \nabla(\nabla^2 I)) \quad (3.6)$$



Rysunek 3.1: Propagacja jasności obrazu wgłąb nieznanego obszaru.

W równaniu (3.6) człon odpowiedzialny za tarcie pomiędzy cząstkami płynu $\nu \nabla^2 V$ został zastąpiony dyfuzją anizotropową, która w kontekście przetwarzania obrazów pierwszy raz wprowadzona została przez Perona i Malika w 1987 roku. Dyfuzja anizotropowa ma znaczny wpływ na zachowanie krawędzi obrazu oraz poprawę jego ostrości. Za dyfuzję anizotropową odpowiada współczynnik g , który przedstawiany jest w różnych postaciach funkcji przewodności. W pracy wykorzystana zostanie postać:

$$g(s) = \frac{1}{1 + \alpha |s|} \quad (3.7)$$

gdzie α , K uprzednio zdefiniowane parametry dyfuzji (stałe), $\nabla^\perp = (-\partial_y, \partial_x)$. Dyskretnie rozwiązanie równania (3.6) jest analogiczne do dyskretnego rozwiązania przepływu płynu w przestrzeni $2D$. W tym celu należy wprowadzić chwile w czasie $t = n \bullet \Delta t$, gdzie $n = \{1, 2, 3, \dots, N\}$, Δt wartość przyrostu czasu pomiędzy dyskretnymi chwilami n i $n + 1$, N moment osiągnięcia stanu ustalonego, matematycznie przedstawianego jako:

$$\frac{\partial I}{\partial t} + \nabla^\perp I \nabla \Delta I \cong 0 \quad (3.8)$$

Warto zauważyć, że człon $\nabla^\perp I$ w równaniu (3.8) odpowiada kierunkowi rozchodzenia się punktów o podobnej jasności, natomiast $\nabla \Delta I$ to kierunek największej zmiany smukłości obrazu. Jeśli wektory są do siebie prostopadłe (do czego zbiega algorytm) iloczyn skalarny wynosi 0. W celu dokładnej analizy dyskretyzacji równanie (3.4) zostanie rozdzielone na dwa podrównania:

$$\frac{d(\nabla^2 I)}{dt} = -(\nabla^\perp I) \cdot \nabla(\nabla^2 I) + \{ad\} \quad (3.9)$$

oraz

$$ad = \nu \nabla(g(\nabla(\nabla^2 I)) \nabla(\nabla^2 I)) \quad (3.10)$$

Rozwiązanie powyższych równań dokonywane będzie na siatce obliczeniowej typu Eulera, a wszystkie wartości wyznaczanych zmiennych będą znajdować się w centrach komórki zdefiniowanej jako konkretny piksel obrazu, zgodnie z Rysunek 3.1 Równanie (3.9) to równanie różniczkowe cząstkowe typu parabolicznego. Zgodnie

	$dx, i - 1$	dx, i	$dx, i + 1$
$dy, j - 1$	I, u, v, w, g	I, u, v, w, g	I, u, v, w, g
dy, j	I, u, v, w, g	I, u, v, w, g	I, u, v, w, g
$dy, j + 1$	I, u, v, w, g	I, u, v, w, g	I, u, v, w, g

Rysunek 3.2: Przykłady obrazów poddanych operacji wmalowywania.

z teorią równanie można rozwiązać metodą różnic skończonych schematem centralnym bądź schematem do przodu. W przypadku równania parabolicznego oba schematy dostarczają stabilnego rozwiązania. W celu ułatwienia dalszych rozważań wprowadzono funkcję f określoną w centrach siatki, z przypisami dolnymi wskazującymi na konkretne miejsce w siatce. Korzystając z rozwinięcia w szereg Taylora gradient funkcji f w punkcie (i, j) można przybliżyć następującym wzorem:

$$\nabla f_{i,j} = \left| \frac{\partial f}{\partial x} \Big|_{i,j}, \frac{\partial f}{\partial y} \Big|_{i,j} \right| \quad (3.11)$$

gdzie:

$$\frac{\partial f}{\partial x} \Big|_{i,j} = \frac{f_{i+1,j} + f_{i-1,j}}{2\Delta x} \quad (3.12)$$

Analogicznie:

$$\frac{\partial f}{\partial y} \Big|_{i,j} = \frac{f_{i,j+1} + f_{i,j-1}}{2\Delta y} \quad (3.13)$$

Na podstawie powyższych wzorów składowe wektora $\nabla^\perp I$, czyli kierunku, w którym należy propagować smukłość obrazu $\nabla^2 I$ wyznacza się z zależności:

$$u_{i,j} = \frac{I_{i,j+1} + I_{i,j-1}}{2\Delta y} \quad (3.14)$$

Oraz

$$v_{i,j} = -\frac{I_{i+1,j} + I_{i-1,j}}{2\Delta x} \quad (3.15)$$

Gdzie:

$$\nabla^\perp I = | -u, v | \quad (3.16)$$

Stosując pięciopunktowy analog różnicowy w metodzie skończonych różnic operator Laplace'a dla przyjętej siatki Eulera można przybliżyć wzorem (przy założeniu $\Delta x = \Delta y$):

$$\begin{aligned} \nabla^2 f_{i,j} &= \Delta f_{i,j} \\ &= \frac{\partial^2 f}{\partial x^2} \Big|_{i,j} + \frac{\partial^2 f}{\partial y^2} \Big|_{i,j} \\ &= \frac{f_{i+1,j+1} + f_{i-1,j+1} + f_{i+1,j-1} + f_{i-1,j-1} - 4f_{i,j}}{\Delta x^2} \end{aligned} \quad (3.17)$$

Korzystając z wyznaczonych powyżej zależności na wartości u oraz v wartość smukłości obrazu $\omega = \nabla^2 I$ można przedstawić w postaci:

$$\Delta I|_{i,j} = \frac{\partial v}{\partial x} \Big|_{i,j} - \frac{\partial u}{\partial y} \Big|_{i,j} \quad (3.18)$$

Podobnie jak w przypadku liczenia pochodnej w przestrzeni powierzchni (obrazu) pierwszą pochodną w przestrzeni czasu przybliża się rozwinięciem w szereg Taylora. Niech przypis górny oznacza konkretną chwilę w czasie. Wtedy wartość pochodnej wynosi:

$$\frac{\partial f}{\partial t} \Big|_{i,j}^n = \frac{f_{i,j}^{n+1} - f_{i,j}^n}{\Delta t} \quad (3.19)$$

Ostatecznie rozwiązanie dyskretnego równania parabolicznego (3.9), gdzie $\Delta x = \Delta y = h$, z wykorzystaniem schematu centralnego ma postać:

$$\begin{aligned} \omega_{i,j}^{n+1} &= \omega_{i,j}^n + \Delta t \left[- \left(\frac{I_{i,j+1} + I_{i,j-1}}{2h} \right) \left(\frac{\omega_{i+1,j} + \omega_{i-1,j}}{2h} \right) \right. \\ &\quad \left. + \left(\frac{I_{i+1,j} + I_{i-1,j}}{2h} \right) \left(\frac{\omega_{i,j+1} + \omega_{i,j-1}}{2h} \right) + \{ad\}_{i,j} \right] \end{aligned} \quad (3.20)$$

Rozwiązanie równania (3.9) można przedstawić również w drugiej postaci wykorzystującej schemat „do przodu”, charakteryzujący się wyznaczaniem wartości pochodnej wirowości w zależności od wyznaczonego kierunku ($\nabla^\perp I$). Wtedy równanie (3.20) przyjmuje postać:

$$\begin{aligned} \omega_{i,j}^{n+1} &= \omega_{i,j}^n + \Delta t \left[- \left(\frac{I_{i,j+1} + I_{i,j-1}}{2h} \right) \left(\frac{\omega_{i+sgn(u^n),j} + \omega_{i,j}}{2h} \right) \right. \\ &\quad \left. + \left(\frac{I_{i+1,j} + I_{i-1,j}}{2h} \right) \left(\frac{\omega_{i,j+sgn(u^n)} + \omega_{i,j}}{2h} \right) + \{ad\}_{i,j} \right] \end{aligned} \quad (3.21)$$

Rozwiązanie równania (3.10), w którym zastąpiono postać zwykłej dyfuzji $\nu \nabla^2 V$ wymaga uważniejszego potraktowania ze względu na większą podatność na niestabilność implementowanych metod dyskretyzacji. Równanie można przedstawić w postaci dywergencji pola wektorowego uzyskanego w wyniku gradientu wirowości z uwzględnionym współczynnikiem dyfuzji anizotropowej:

$$\begin{aligned}\{ad\} &= \nu \nabla (g(\nabla(\nabla^2 I)) \nabla(\nabla^2 I)) \\ &= \nu \nabla \cdot (g(|\nabla \omega|) \nabla \omega) \\ &= \partial_x (g(|\nabla \omega|) \partial_x \omega) + \partial_y (g(|\nabla \omega|) \partial_y \omega)\end{aligned}\tag{3.22}$$

(3.22) gdzie: $\partial_x = \frac{\partial}{\partial x}$ oraz analogicznie: $\partial_y = \frac{\partial}{\partial y}$. Zgodnie z [12] powyższe równanie można rozwiązać w poniższych krokach: Korzystając z zależności (3.12) i (3.13) wyznaczyć wartości $(\frac{d\omega}{dx})_{i,j}$ oraz $(\frac{d\omega}{dy})_{i,j}$. Na podstawie wartości z punktu 1 obliczyć moduł wektora dla przestrzeni Euklidesowej:

$$|\nabla \omega|_{i,j} = \sqrt{\left(\frac{\partial \omega}{\partial x}\right)_{i,j}^2 + \left(\frac{\partial \omega}{\partial y}\right)_{i,j}^2}\tag{3.23}$$

Na podstawie wartości z punktu 2 wyznaczyć wartość

$$g(|\nabla \omega|_{i,j})_{i,j}$$

wykorzystując (3.6). Korzystając z wartości wyznaczonych powyżej oraz ponownie z zależności (3.12) i (3.13) obliczyć:

$$\begin{aligned}\{ad\}_{i,j} &= \left(\partial_x \left(g(|\nabla \omega|_{i,j})_{i,j} \left(\frac{d\omega}{dx} \right)_{i,j} \right) \right)_{i,j} \\ &+ \left(\partial_y \left(g(|\nabla \omega|_{i,j})_{i,j} \left(\frac{d\omega}{dy} \right)_{i,j} \right) \right)_{i,j}\end{aligned}\tag{3.24}$$

Innym sposobem dyskretyzacji równania (3.10) może być metoda przedstawiona w [13], wykorzystująca asymetryczny schemat liczenia pierwszej pochodnej funkcji. Pierwsze pochodne asymetryczne lewe przyjmują postać:

$$\frac{\partial^- f}{\partial x} \Big|_{i,j} = \frac{f_{i,j} + f_{i-1,j}}{\Delta x}\tag{3.25}$$

$$\left. \frac{\partial^- f}{\partial y} \right|_{i,j} = \frac{f_{i,j} + f_{i,j-1}}{\Delta y} \quad (3.26)$$

Analogicznie prawe:

$$\left. \frac{\partial^+ f}{\partial x} \right|_{i,j} = \frac{f_{i+1,j} + f_{i,j}}{\Delta x} \quad (3.27)$$

$$\left. \frac{\partial^+ f}{\partial y} \right|_{i,j} = \frac{f_{i,j+1} + f_{i,j}}{\Delta y} \quad (3.28)$$

Następnie powyższe zależności znajdują zastosowanie w wyznaczeniu członów $\partial_y (g(|\nabla\omega|) \partial_y \omega)$ oraz $\partial_x (g(|\nabla\omega|) \partial_x \omega) + \partial_y$. Stosując naprzemiennie schemat lewy z prawym do liczenia wartości pochodnych wewnętrznych i zewnętrznych w wyniku otrzymuje się ogólny schemat centralny. Stosując powyższe ostatecznie równanie (3.10) można przedstawić w następującej postaci dyskretnej:

$$\begin{aligned} \{ad\}_{i,j} = \frac{1}{2} & \left[(g_{i,j} + g_{i+1,j}) (\omega_{i+1,j} - \omega_{i,j}) \right. \\ & - (g_{i-1,j} + g_{i,j}) (\omega_{i,j} - \omega_{i-1,j}) \\ & + (g_{i,j} + g_{i,j+1}) (\omega_{i,j+1} - \omega_{i,j}) \\ & \left. - (g_{i,j-1} + g_{i,j}) (\omega_{i,j} - \omega_{i,j-1}) \right] \end{aligned} \quad (3.29)$$

Należy zauważyć, iż do wyznaczenia wartości u, v oraz $\nabla^2 I$ wykorzystywanych do obliczeń (3.9) i (3.10) w każdej kolejnej chwili czasu n , konieczna jest znajomość obrazu I^n . W związku z tym równoległe do równania (3.5) należy rozwiązywać poniższe niejednorodne równanie różniczkowe cząstkowe liniowe drugiego rzędu.

$$\Delta I^n = \omega^n \quad (3.30)$$

Dla przestrzeni dwuwymiarowej równanie przyjmuje postać:

$$\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} = \omega \quad (3.31)$$

W celu przybliżenia drugiej pochodnej funkcji f w punkcie (i, j) należy rozwinąć funkcję f w dwa szeregi Taylora względem punktu $(i+1, j)$ oraz $(i-1, j)$. Po rozwiązaniu sumy dwóch rozwinięć względem $\frac{\partial^2 f}{\partial x^2}$ otrzymuje się poniższy sposób dyskretyzacji drugiej pochodnej:

$$\left. \frac{\partial^2 f}{\partial x^2} \right|_{i,j} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2} \quad (3.32)$$

Analogicznie:

$$\left. \frac{\partial^2 f}{\partial y^2} \right|_{i,j} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta y^2} \quad (3.33)$$

Korzystając z (3.32) oraz (3.33) równanie (3.30) w punkcie (i, j) przyjmuje postać pięciopunktowego analogu różnicowego operatora Δ ($\Delta x = \Delta y = h$):

$$I_{i+1,j} + I_{i-1,j} + I_{i,j+1} + I_{i,j-1} - 4I_{i,j} = h^2 \omega_{i,j} \quad (3.34)$$

Na podstawie równania (3.34) można stworzyć równanie macierzowe postaci $Ax = B$, gdzie x reprezentuje obraz I , B smukłość obrazu, natomiast A to macierz charakterystyczna współczynników. Istnieje wiele metod rozwiązywania równań Poissona takich jak metoda Jacobiego czy Gaussa-Seidla. Wystarczającą wydajność i szybkość działania w celach rozwiązywania równania można uzyskać stosując metodę SOR – sukcesywnej nadrelaksacji. Dyskretyzacja równania wygląda następująco:

$$I_{i,j}^{n+1} = (1 - \beta) I_{i,j}^n + \beta \cdot \frac{1}{2} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1} \cdot \left[\frac{1}{\Delta x^2} (I_{i+1,j}^n + I_{i-1,j}^{n+1}) + \frac{1}{\Delta y^2} (I_{i,j+1}^n + I_{i,j-1}^{n+1}) + \omega_{i,j}^n \right] \quad (3.35)$$

Metoda w swoim równaniu przyjmuje parametr wagowy β odpowiadający za zbieżność algorytmu. Dobór nie jest losowy i zawiera się w przedziale $(0, 2)$. Zgodnie z literaturą parametr ten powinno przyjmować się zgodnie z zależnością:

$$\omega_{opt} = 2 \left(1 + \frac{\pi}{N} \right)^{-1} \quad (3.36)$$

gdzie N to liczba kolumn przyjętej siatki obliczeniowej.

Operacje wmalowywania bardzo często wykonywane są w punktach maski tworzących nieregularne kształty, mogących znajdować się w różnych częściach obrazu. Stąd powyżej przedstawiony schemat obliczeń wykonywany jest dla wszystkich pikseli obrazu. Po każdej iteracji uzyskane wartości I^{n+1} w znanych punktach przepisywane są z oryginalnego obrazu I_o , zgodnie z zależnością poniżej.

$$I^{n+1}(D/\Omega) = I_o(D/\Omega) \quad (3.37)$$

Podsumowując w celu wyznaczenia niewiadomych wartości obrazu w punktach zdefiniowanych przez maskę należy powtarzać następujące iteracje do uzyskania stanu ustalonego:

1. Rozwiązać równanie (3.24)
2. Na podstawie równania (3.20) zamiennie z (3.21) wyznaczyć wartości $\omega_{i,j}^{n+1}$ dla każdego punktu (i, j)
3. Korzystając z (3.35) wyznaczyć jasność każdego piksela $I_{i,j}^{n+1}$.
4. Do uzyskanego obrazu w punkcie 3 przepisać znane wartości z obrazu oryginalnego

Powyższy algorytm przedstawiono dla obrazów w odcieniach szarości. W celu implementacji rozwiązania dla obrazów kolorowych, których funkcję definiuje się jako przekształcenie w pole wektorowe $[R, G, B]$, czyli $I : D \rightarrow R^3$ zgodnie z propozycją w [14] można w pierwszym kroku przekształcić do współrzędnych sferycznych $[I_1, I_2, I_3]$ korzystając z zależności:

$$I_{1,i,j} = r_{i,j} = \sqrt{R_{i,j}^2 + G_{i,j}^2 + B_{i,j}^2} \quad (3.38)$$

$$I_{2,i,j} = \sin\Theta_{i,j} = \sin\theta_{i,j}^{-1} \frac{G_{i,j}}{R_{i,j}} \quad (3.39)$$

$$I_{3,i,j} = \sin\phi_{i,j} = \sin\theta_{i,j}^{-1} \frac{B_{i,j}}{r_{i,j}} \quad (3.40)$$

Następnie dysponując komponentami I_1, I_2, I_3 wyznaczonymi z powyższych zależności, dla każdego z nich oddzielnie należy wykonać wcześniej opisane iteracje. W celu odwrotnego przekształcenia $[I_1, I_2, I_3] \rightarrow [R, G, B]$ wystarczy wykonać poniższe obliczenia:

$$B_{i,j} = r_{i,j} \cos\Theta_{i,j} \cos\phi_{i,j} \quad (3.41)$$

$$R_{i,j} = r_{i,j} \cos\Theta_{i,j} \sin\phi_{i,j} \quad (3.42)$$

$$G_{i,j} = r_{i,j} \sin\theta_{i,j} \quad (3.43)$$

Operację wmalowywania można wykonać na pierwotnych komponentach $[R, G, B]$ lecz matematyczna transformacja przestrzeni charakteryzuje się lepszymi właściwościami związanymi z postrzeganiem różnicy barw δE przez ludzkie oko:

$$\Delta E = \sqrt{(\Delta I_1)^2 + (\Delta I_2)^2 + (\Delta I_3)^2} \quad (3.44)$$

3.2 Problem minimalizacji energii

Innym sposobem zastosowania równań różniczkowych w procesie wmalowywania obrazu może być adaptacja problemu minimalizacji energii. Pojęcie wahanja funkcji TV w analizie obrazów pierwszy raz wprowadzone zostało przez Rudin, Osher i Fatemi'ego w [15], gdzie proponowany model znajduje zastosowanie w usuwaniu szumów z obrazu. Następnie model TV zostaje zaproponowany do wmalowywania obrazu w [4] przez Chan i Shen'a. Problem doboru parametrów dla modelu TV , właściwości oraz proces dyskretyzacji problemu minimalizacji jest dokładnie przedstawiony w [16]. Algorytm bazujący na lokalnych funkcjonalach TV charakteryzuje się zdolnością do kontynuacji geometrycznych struktur obrazu i krawędzi. Zgodnie z przedstawioną literaturą wadą algorytmu jest brak możliwości propagacji tekstury w nieznany obszar Ω . Warto wspomnieć, iż w przypadku usuwania szumów z obrazów wykorzystanie lokalnych operatorów nie pozwala zachować tekstur. Problem jednoczesnej propagacji geometrii i tekstury można rozwiązać przy zastosowaniu nielokalnego modelu, który zostanie poddany analizie w niniejszej pracy. Klasyczny nielokalny model wahanja funkcji został intensywnie przebadany dla obrazów w odcieniach szarości. W przypadku obrazów kolorowych jednym z pierwszych zaproponowanych modeli jest model Mumford-Shah dokładnie opisany w [17]. Model ten dla każdej z warstw obrazu kolorowego przeprowadza oddzielną analizę, w wyniku prowadząc do niejednorodnego kontrastu kolorów obrazu w przestrzeni wmalowywanej Ω z jego otoczeniem D/Ω . Problem uzyskania jednokolorowego kontrastu rozwiązano poprzez propozycję nowego, szybkiego modelu w [18] wprowadzającego zależność pomiędzy wszystkimi warstwami obrazu kolorowego. W celu analizy nowego, nielokalnego modelu wahanja funkcji dla obrazów kolorowych $NL - CTV$ zaproponowanego w [18] należy uprzednio wprowadzić oznaczenia i zależności nielokalnych operatorów. Zgodnie z poprzednim działem niech D stanowi dziedzinę obrazu. Wtedy obraz w odcieniach szarości możemy przedstawić jako przekształcenie $I(x) : D \rightarrow R$, $x \in R$ zdefiniowaną dla każdego piksela x ze zbioru D . Wtedy pojęcie nielokalnego gradientu pomiędzy pikselem x i y , gdzie

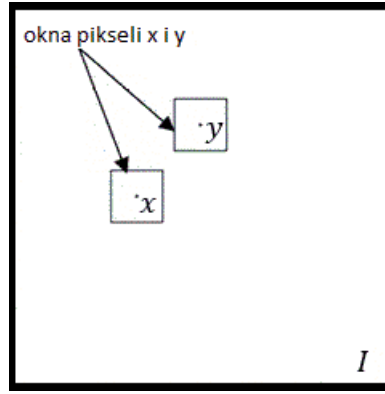
$x, y \in D$ z definicji przedstawia się zależnością:

$$\nabla_{NL} I(x, y) \triangleq [I(y) - I(x)] \sqrt{w(x, y)} \quad (3.45)$$

Warto zauważyć, że wynikiem powyższej zależności nie jest wektor, a wartość skalarna. Wyrażenie stanowi proste mapowanie $D \times D \rightarrow R$. Wartość wagi $w(x, y)$ z definicji wyraża się poniższą zależnością:

$$w(x, y) \triangleq \exp \left\{ -\frac{G_\sigma * |I(x + \cdot) - I(y + \cdot)|^2}{r^2} \right\} \quad (3.46)$$

Wartość r to uprzednio definiowana stała skalująca wartości podobieństwa pomie-



Rysunek 3.3: Kształt okien wykorzystywanych do obliczania wartości $w(x, y)$.

dzy pikselami. Podobieństwo wyznaczane jest na podstawie utworzonych okien scentrowanych względem piksela x oraz y oznaczanych operatorem "." zgodnie z Rysunek 3.3. Normę macierzową występującą w powyższej zależności można wyznaczyć korzystając z normy Frobeniusa:

$$|I(x + \cdot) - I(y + \cdot)|^2 = \sum_{i=1}^n \sum_{j=1}^n (I_{x+ \cdot}(i, j) - I_{y+ \cdot}(i, j))^2 \quad (3.47)$$

gdzie wartość n odpowiada wielkości badanego okna i stanowi uprzednio definiowany parametr. Zmienna G_σ reprezentuje macierz o rozmiarze zgodnym z operatorem otoczenia danego piksela ".". Wartości wewnątrz macierzy pozwalają stopniować wpływ każdego z rozpatrywanych pikseli na uzyskiwany wynik poprzez przyznawanie większego priorytetu pikselom centralnym, przy deprecjonowaniu tych

bardziej oddalonych od centrum. Podobnie jak w przypadku nielokalnego gradientu funkcja wagi stanowi mapowanie $DxD \rightarrow R$. Kolejnymi ważnymi nielokalnymi operatorami wykonanymi w punkcie x są:

- iloczyn skalarny dwóch nielokalnych mapowań $\vec{p} : DxD \rightarrow R$ (mapowaniem \vec{p} może być wspomniana funkcja wagi, bądź nielokalny gradient):

$$(\vec{p}_1 \cdot \vec{p}_2)(x) \triangleq \int_D p_1(x, y) \cdot p_2(x, y) dy \quad (3.48)$$

- moduł mapowania \vec{p} :

$$|\vec{p}|(x) \triangleq \sqrt{\vec{p} \cdot \vec{p}} = \sqrt{\int_D (p(x, y))^2 dy} \quad (3.49)$$

- dywergencja mapowania \vec{p} :

$$(\nabla_{NL} \cdot \vec{p})(x) \triangleq \int_D (p(x, y) - p(y, x)) \sqrt{w(x, y)} dy \quad (3.50)$$

- laplasjan mapowania

$$\Delta_{NL} \vec{p}(x) \triangleq \frac{1}{2} \nabla_{NL} \cdot (\nabla_{NL} I(x)) = \int_D (I(y) - I(x)) w(x, y) dy \quad (3.51)$$

Warto zauważyć, iż w przypadku mapowań w postaci nielokalnego gradientu (3.45) oraz wagi (3.46) zachodzą odpowiednio zależności:

$$\begin{aligned} (p(x, y) - p(y, x)) &= (\nabla_{NL} I(x, y) - \nabla_{NL} I(y, x)) \\ &= (\nabla_{NL} I(x, y) + \nabla_{NL} I(x, y)) \\ &= 2\nabla_{NL} I(x, y) \end{aligned} \quad (3.52)$$

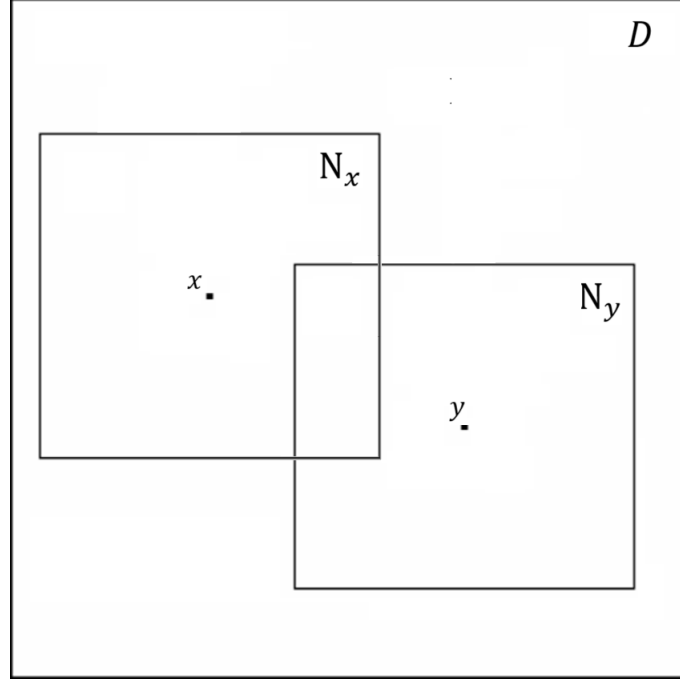
Stąd w szczególnym przypadku nielokalną dywergencję można przedstawić:

$$(\nabla_{NL} \cdot \vec{p})(x) \triangleq \int_D 2(p(x, y)) \sqrt{w(x, y)} dy \quad (3.53)$$

$$w(x, y) = w(y, x) \quad (3.54)$$

Nielokalna dywergencja stanowi mapowanie $DxD \rightarrow D$. W celu dyskretyzacji powyższych równań należy zgodnie z [19] wprowadzić pojęcie okna poszukiwania N_i ,

gdzie $i \in D$, $j \in N_i$. Rozmiar okna stanowi uprzednio definiowany parametr. Dla wszystkich pikseli w oknie zachodzi zależność $N_i := \{j : w_{i,j} > 0\}$. Wprowadzenie okna skutkuje ograniczeniem wcześniej wprowadzonego mapowania $D \times D \rightarrow R$ do $N \times D \rightarrow R$.



Rysunek 3.4: Przykładowe okna N_x, N_y utworzone dla pikseli x, y .

Niech przypisy dolne wskazują na dyskretną lokalizację w przestrzeni obrazu D . Wtedy nielocalne operatory przedstawić można za pomocą zdyskretyzowanych równań:

$$\nabla_{NLd} I_{i,j} \triangleq (I_j - I_i) \sqrt{w_{i,j}} \quad (3.55)$$

$$w_{i,j} = \exp \left\{ - \sum_{l \in G_\sigma, j \in N_i, k \in N_j} G_{\sigma l} * |I_j - I_k|^2 \right\} \quad (3.56)$$

$$|\vec{p}|_i \triangleq \sqrt{\sum_{j \in N_i} (p_{i,j})^2} \quad (3.57)$$

$$\nabla_{NLd} \cdot \vec{p}_i \triangleq \sum_{j \in N_i} (p_{i,j} - p_{j,i}) \sqrt{w_{i,j}} \quad (3.58)$$

$$\Delta_{NL} \vec{p}_i \triangleq \sum_{j \in N_i} (I_j - I_i) w_{i,j} \quad (3.59)$$

W szczególnym przypadku dla (3.53) można zapisać:

$$\nabla_{NLd} \cdot \vec{p}_i \triangleq \sum_{j \in N_i} 2p_{i,j} \sqrt{w_{i,j}} \quad (3.60)$$

Warto zauważyć, iż indeksy i, j wskazują na konkretny piksel w obrazie. Lokalizacji w przestrzeni $2D$ dokonuje się przez znajomość par: $i = (x_i, y_i)$ oraz $j = (x_j, y_j)$, gdzie x oś rzędnych, odpowiednio y oś odciętych.

Nielokalny model wahanía funkcji oznaczany $NL-TV$ definiowany dla obrazów w odcieniach szarości przyjmuje następującą postać:

$$\min_u \left\{ E(u) = \int_D |\nabla_{NL} u(x)| dx + \frac{1}{2} \int_D \lambda_d (u - f)^2 dx \right\} \quad (3.61)$$

gdzie f stanowi oryginalny obraz wejściowy $f(x) : D \rightarrow \mathbb{R}$, natomiast $u(x) : D \rightarrow \mathbb{R}$ obraz wynikowy, będący obrazem wmalowanym, bądź w przypadku pierwotnych założeń obrazem z usuniętym szumem. W przypadku wmalowywania zmienna λ_d to funkcja definiowana na podstawie maski obrazu przyjmująca następujące wartości:

$$\lambda_d(x) = \begin{cases} 0 & x \in \Omega \\ 1 & x \in D/\Omega \end{cases} \quad (3.62)$$

Tak zdefiniowany model można przekształcić wykorzystując operację rozbicia Bregman'a. Rozbicie stosuje się w celu poprawienia wydajności obliczeniowej w przypadku dyskretnej implementacji problemu minimalizacji energii. Wprowadzenie pomocniczej zmiennej $\vec{v} = v(x, y) : \Omega \times \Omega \rightarrow \mathbb{R}$ i parametru Bregman'a $\vec{b} = \vec{b}(x, y) : \Omega \times \Omega \rightarrow \mathbb{R}$ pozwala przekształcić problem minimalizacji powyższej energii do następującej postaci iteracyjnej:

$$\begin{aligned} (u^{k+1}, \vec{v}^{k+1}) &= \arg \min_{u, \vec{v}} E(u, \vec{v}) \\ &= \left\{ \arg \min_{u, \vec{v}} \int_D |\vec{v}|(x) dx + \frac{1}{2} \int_D \lambda_d (u - f)^2 dx \right. \\ &\quad \left. + \frac{\Theta}{2} \int_D \left| \vec{v} - \nabla_{NL} u - \vec{b}^{k+1} \right|^2(x) dx \right\} \end{aligned} \quad (3.63)$$

$$\vec{b}^{k+1} = \vec{b}^k + \nabla_{NL} u^k - \vec{v}^k, \vec{b}^0 = \vec{v}^0 = 0 \quad (3.64)$$

gdzie Θ stanowi uprzednio definiowany parametr algorytmu. Stosując podstawowe równanie rachunku wariacyjnego postaci równania Eulera-Lagrange'a, metodę przedstawioną w [20] oraz schemat iteracyjny Gauss'a-Seidel'a powyższe równania można sprowadzić do problemu iteracyjnego rozwiązania poniższych zależności:

$$\lambda_D (u - f) + \Theta \nabla_{NL} \cdot \left(\vec{v}^k - \nabla_{NL} u - \vec{b}^{k+1} \right) = 0 \quad (3.65)$$

$$\vec{v}^{k+1} = \max \left(\left| \nabla_{NL} u^{k+1} + \vec{v}^{k+1} \right| - \frac{1}{\Theta}, 0 \right) \cdot \frac{\nabla_{NL} u^{k+1} + \vec{b}^{k+1}}{\left| \nabla_{NL} u^{k+1} + \vec{b}^{k+1} \right|} \quad (3.66)$$

Zależności można stosować dla obrazów w odcieniach szarości, bądź obrazów kolorowych traktując każdą z poszczególnych warstw osobnym algorytmem iteracyjnym. Zgodnie z [18] takie rozwiązanie prowadzi do niezachowania odpowiedniego kontrastu odrestaurowanej części obrazu i niezachowania krawędzi.

W celu uzależnienia od siebie wszystkich warstw obrazu autorzy w [18] proponują zastosowanie funkcjonału MTV przedstawionego w [21], funkcjonału CTV przedstawionego w [22], bądź model Mumford-Shah'a dokładnie opisanego w [17]. Ze względu na wydajniejszą implementację oraz najlepsze wyniki uzyskiwane w przypadku modelu CTV został on wybrany do badań w niniejszej pracy. Wykorzystując wspomniany funkcjonał energia przyjmuje postać:

$$\min_u \left\{ E(u) = \sqrt{\sum_{i=1}^m \left(\int_D |\nabla_{NL} u(x)| dx \right)^2} + \frac{1}{2} \sum_{i=1}^m \int_D \lambda_d (u_i - f_i)^2 dx \right\} \quad (3.67)$$

gdzie m odpowiada ilości warstw tworzących kolorowy obraz. Zastosowanie równania rachunku wariacyjnego postaci Eulera-Lagrange'a w swoim rozwiązaniu prowadzi do konieczności obliczenia nielokalnej krzywizny krzywej, będącej w przypadku dyskretyzacji rozwiązania znacznie obciążającą obliczeniowo operacją. W tym celu, podobnie jak w przypadku poprzedniego modelu $NL-TV$ zastosowany zostanie heurystyczny algorytm rozbicia Bregman'a. Analogicznie wprowadzone zostają zmienne $\vec{v} = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_m)$ oraz $\vec{b} = (\vec{b}_1, \vec{b}_2, \dots, \vec{b}_m)$. Wtedy przyto-

czona energia przyjmuje iteracyjną postać:

$$\begin{aligned} \min_u E(u) = \min_u \left\{ \sqrt{\sum_{i=1}^m \left(\int_D |\vec{v}_i| (x) dx \right)^2} \right. \\ \left. + \frac{1}{2} \sum_{i=1}^m \int_D \lambda_d (u_i - f_i)^2 dx + \frac{\theta}{2} \sum_{i=1}^m \int_D \left| \vec{v}_i - \nabla_{NL} u_i - \vec{b}_i^{k+1} \right|^2 (x) dx \right\} \end{aligned} \quad (3.68)$$

$$\vec{b}_i^{k+1} = \vec{b}_i^k + \nabla_{NL} u_i^k - \vec{v}_i^k, \quad \vec{b}_i^0 = \vec{v}_i^0 = 0 \quad (3.69)$$

Przyjmując naprzemienną strategię minimalizacji energii można uzyskać równania Eulera-Lagrange'a wyznaczone oddzielnie względem zmiennych u oraz v :

$$\lambda_D (u_i - f_i) + \Theta \nabla_{NL} \cdot \left(\vec{v}_i^k - \nabla_{NL} u_i - \vec{b}_i^{k+1} \right) = 0 \quad (3.70)$$

$$\Theta \left(\vec{v}_i - \nabla_{NL} u_i^{k+1} - \vec{b}_i^{k+1} \right) + \frac{\int_D |\vec{v}_i| (x) dx}{\sqrt{\sum_{i=1}^m \left(\int_D |\vec{v}_i| (x) dx \right)^2}} \frac{\vec{v}_i}{|\vec{v}_i| (x)} = 0 \quad (3.71)$$

W celu przedstawienia dyskretnej formy równań (3.70) i (3.71) i (3.71) wygodnym będzie zapis dla konkretnego piksela $l \in D$ w obrazie. W przypadku (3.70):

$$(\lambda_D)_l [(u_i)_l - (f_i)_l] + \Theta \nabla_{NL} \cdot \left[\left(\vec{v}_i^k \right)_l - (\nabla_{NL} u_i)_l - \left(\vec{b}_i^{k+1} \right)_l \right] = 0 \quad (3.72)$$

Korzystając z dyskretnych form (3.45), (3.46), (3.48) i (3.50) równanie (3.70) można przedstawić w postaci:

$$\begin{aligned} (u_i)_l^{k+1} = \frac{1}{(\lambda_D)_l + 2\Theta \sum_{j \in N_i} (w_i)_{l,j}} \left[2\Theta \sum_{j \in N_i} (u_i^k)_j (w_i)_{l,j} \right. \\ \left. + (\lambda_D)_l (f_i)_l \right. \\ \left. - \Theta \left(\sum_{j \in N_i} \left((v_i^k)_{l,j} - ((v_i^k)_{l,j} - ((v_i^k)_{l,j} + ((v_i^k)_{l,j}) \sqrt{(w_i)_{l,j}} \right) \right] \right) \end{aligned} \quad (3.73)$$

W przypadku równania (3.70) Eulera-Lagrange'a należy przekształcić je względem

zmiennej \vec{v} :

$$\vec{v}_i^{k+1} = \max \left(\left| \nabla_{NL} u_i^{k+1} + \vec{b}_i^{k+1} \right| - \frac{\int_D |\vec{v}_i| (x) dx}{\Theta \sqrt{\sum_{i=1}^m \left(\int_D |\vec{v}_i^k| (x) dx \right)^2}}, 0 \right) \cdot \frac{\nabla_{NL} u_i^{k+1} + \vec{b}_i^{k+1}}{\left| \nabla_{NL} u_i^{k+1} + \vec{b}_i^{k+1} \right|} \quad (3.74)$$

Podobnie, korzystając z dyskretnych form (3.45), (3.46), (3.47), (3.48) i (3.50) równanie (3.74) można przedstawić w postaci:

$$\left(\vec{v}_i^{k+1} \right)_l \cong \max \left(|A_i^{k+1}|_l - \frac{B_i^k}{\Theta \sqrt{\sum_{i=1}^m (B_i^k)^2}}, 0 \right) \frac{(A_i^{k+1})_l}{|A_i^{k+1}|_l} \quad (3.75)$$

gdzie:

$$(A_i^{k+1})_l = \left((u_i^{k+1})_j - (u_i^{k+1})_l \right) \sqrt{(w_i)_{l,j}} + (b_i^{k+1})_{l,j} \quad (3.76)$$

$$|A_i^{k+1}|_l = \sqrt{\sum_j \left(\left((u_i^{k+1})_j - (u_i^{k+1})_l \right) \sqrt{(w_i)_{l,j}} + (b_i^{k+1})_{l,j} \right)^2} \quad (3.77)$$

$$B_i^k = \sum_l \sqrt{\sum_j (v_i^k)_{l,j}^2} \quad (3.78)$$

Warto przypomnieć, iż w przytoczonych wzorach j odpowiada pikselom znajdującym się oknie poszukiwania N_l scentrowanym względem piksela l , a i odpowiada i -tej warstwie spośród warstw tworzących obraz kolorowy. Autorzy w [17] proponują zmodyfikowany sposób wyznaczania wagi uwzględniający kształt maski w obrazie:

$$w(x, y) \triangleq \exp \left\{ - \frac{G_\sigma * \chi_R(x + \cdot) |I(x + \cdot) - I(y + \cdot)|^2}{r^2} \right\} \quad (3.79)$$

gdzie:

$$\chi_d(x) = \begin{cases} 0 & x \in \Omega \\ 1 & x \in D/\Omega \end{cases} \quad (3.80)$$

Podsumowując algorytm wmalowywania w oparciu o model $NL - CTV$ z algorytmem rozbicia Bregman'a można przedstawić w następujących krokach:

1. Inicjalizacja zmiennych $k = 0$, $\vec{b}_i^0 = 0$, $\vec{v}_i^0 = 0$, $u_i^0 = f_i^0$
2. Wyznaczenie wagi w_i^2 zgodnie z (3.79) dla całego obszaru obrazu D
3. Powtarzanie do spełnienia kryterium końcowego:
 - Wyznaczenie $\vec{b}^{k+1} = \vec{b}^k + \nabla_{NL} u^k - \vec{v}^k$
 - Wyznaczenie u^{k+1} na podstawie (3.72) dla każdego piksela l
 - Wyznaczenie v^{k+1} na podstawie (3.75) dla każdego piksela l
4. Aktualizacja wagi w_i^2 w obszarze Ω

Kryterium definiującym zakończenie algorytmu może być uzyskanie granicznej wartości różnicy energii ϵ uprzednio definiowanej pomiędzy wykonywanymi iteracjami:

$$|E^{k+1} - E^k| \leq \epsilon \quad (3.81)$$

3.2.1 Modyfikacje algorytmu

Modyfikacja funkcji wagi.

Podczas testowania opisanego powyżej algorytmu wmalowywania jedną z wad okazała się zbyt lokalna analiza obrazu, powodująca brak odpowiedniej kontynuacji struktury. Zwiększając parametry w postaci rozmiaru skrawka i rozmiaru okna algorytm wprowadza zbytne uśrednienie nie pozwalające odpowiednio kontynuować tekstury obrazu. W celu uniknięcia zjawiska rozmycia tekstur, przy jednoczesnej poprawie zdolności algorytmu do kontynuacji większych struktur zaproponowano modyfikację funkcji wagi (3.46) do następującej postaci:

$$w(x, y) = \exp \left\{ -\frac{G_\sigma * |I(x + \cdot) - I(y + \cdot)|^2}{r^2} \right\} \cdot \exp \left\{ -\frac{G_\sigma * |I(x + s(\cdot)) - I(y + s(\cdot))|^2}{r^2} \right\} \quad (3.82)$$

W powyższej zależności człon $s(\cdot)$ oznacza przeskalowanie domyślnej wartości przyjętego rozmiaru skrawka do rozmiaru s -krotnie większego. Edycja nie dotyczy pozostałej części algorytmu i wpływa jedynie na wartość wagi.

Modyfikacja wyznaczania wartości obrazu u .

Złożoność obliczeniowa algorytmu skutkuje w względnie długim czasie wypełniania obrazu. W celu skrócenia czasu wyznaczania wynikowego obrazu zastosowano następującą aproksymację równania (3.73) pozwalającą uzyskać przybliżone wyniki do oryginalnego algorytmu:

$$(u_i)_l^{k+1} = \frac{1}{(\lambda_D)_l + 2\Theta \sum_{j \in N_i} (w_i)_{l,j}} \left[2\Theta \sum_{j \in N_i} (u_i^k)_j (w_i)_{l,j} + (\lambda_D)_l (f_i)_l \right] \quad (3.83)$$

W powyższym równaniu eliminacja zmiennych v oraz b pozwala znacznie ograniczyć potrzeby pamięciowe programu oraz przyspieszyć działanie dzięki pominięciu kroków, w których wyznaczane są przytoczone zmienne.

Połączenie metody Criminisi z funkcją wagi.

Kolejną z pozytywnych prób modyfikacji algorytmu NLCTV w pracy okazała się próba zastąpienia problemu minimalizacji energii (3.67) opartej na funkcji wagi, metodą syntezy obrazu (bazującej na metodzie syntezy tekstury opisanej w dalszej części pracy w Rozdział 5). Kroki nowego algorytmu można przedstawić w następującym schemacie:

1. Ustawienie argumentów algorytmu: obraz u^0 , maska obrazu χ , wielkość okna SW , wielkość skrawka obrazu PS
2. Wyznaczenie funkcji wagi w zgodnie z (3.56) dla parametrów SW i PS i χ
3. Powtarzanie do uzyskania zbioru $\Omega = \emptyset$
 - a) Znalezienie dla każdego piksela $p \in \partial\Omega$ najlepszego dopasowania $d \in D \cap N_p$, gdzie N_p zbiór pikseli z okna utworzonego dla piksela p , dla którego zachodzi:

$$d = \arg \max_{t \in D \cap N_p} w(p, t) \quad (3.84)$$

a następnie przepisanie:

$$u(p) = u(d) \quad (3.85)$$

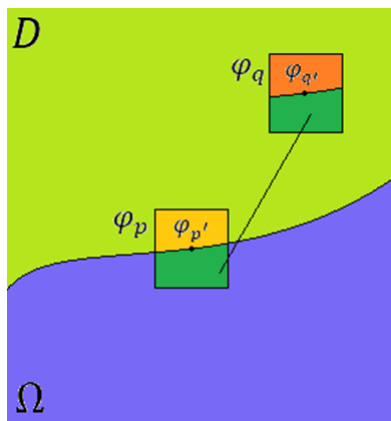
b) Pomniejszenie maski Ω o granicę $\partial\Omega$

Zastosowana modyfikacja pozwala na szybkie wyznaczenie wartości obrazu w szukanym obszarze Ω . Przedstawiony sposób pozwala uniknąć sytuacji, w której tworzony jest obcy kolor będący z poza gamy dostępnych kolorów w obrazie. Taki sposób wyznaczania wpływa na wartość wag obliczanych w kolejnych iteracjach i niweluje błąd wyznaczania wartości z poza dostępnej gamy w kolejnych iteracjach dla $\partial\Omega$. Podobnie jak dla modelu NLCTV można zastosować modyfikację funkcji wagi w zgodnie z (3.82), w celu uzyskania podobnego efektu.

Rozdział 4

Uzupełnianie kawałkami obrazu.

Kolejną grupą algorytmów rozważanych w niniejszej pracy jest grupa opierająca się na wypełnianiu brakujących obszarów w obrazie skrawkami pochodzącymi z ich otoczenia. W odróżnieniu od metod bazujących na równaniach różniczkowych metoda uzupełniania braków nie bazuje na metodzie dyfuzji. Pierwszy raz zaproponowana została w [6], a jej podstawowy krok opiera się na schemacie przedstawionym na poniższym rysunku. Zgodnie z Rysunek 4.1 niech zbiór D oznaczony kolorem



Rysunek 4.1: Schemat wmalowywania obrazu.

jasnoniebieskim określa znane punkty obrazu, natomiast ω oznaczony kolorem czerwonym określa obszar do odrestaurowania. Niech obraz wejściowy stanowi funkcję Ω . Niech φ_p stanowi podzbiór punktów tworzących skrawek o uprzednio zdefiniowanej wielkości, scentrowany względem piksela p znajdującego się na granicy maski

$\partial\Omega$. Niech φ_q stanowi podzbiór punktów tworzących skrawek o rozmiarze równym rozmiarowi φ_p , scentrowany względem piksela q dla którego żaden z punktów wyznaczonego skrawka nie znajduje się w obszarze maski. Niech $\varphi_{p'}$ oznaczone kolorem żółtym stanowi podzbiór punktów znajdujących się w φ_p , które nie znajdują się w obrębie maski. Wtedy $\varphi_{q'}$ oznaczone kolorem pomarańczowym stanowi grupę pikseli zlokalizowanych odpowiednio według pikseli z podzbioru $\varphi_{p'}$. Zgodnie z nomenklaturą matematyczną:

$$p \in \partial\Omega \quad (4.1)$$

$$q \in \phi \quad (4.2)$$

$$\phi \subset D \quad (4.3)$$

$$\varphi_p = (p + \cdot) \quad (4.4)$$

$$\varphi_q = (q + \cdot) \quad (4.5)$$

$$\varphi_{p'} = \varphi_p \cap (D \cup W) \quad (4.6)$$

$$\varphi_{q'} \subset \varphi_q \quad (4.7)$$

gdzie W przedstawia zbiór wmalowanych pikseli uprzednio znajdujących się w rejonie maski.

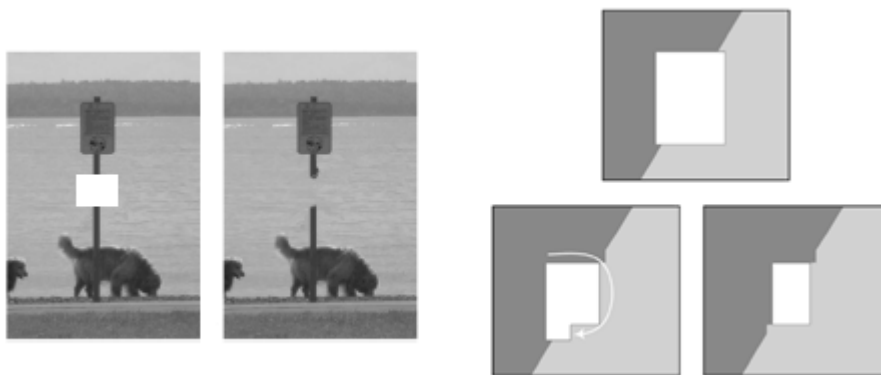
Podstawową operacją wmalowywania metodą propagacji tekstury obrazu jest znalezienie najpodobniejszej pary $\langle \varphi_{p'}, \varphi_{q''} \rangle$ oraz zgodnie z powyższym rysunkiem przepisanie wartości odpowiednich pikseli z wyznaczonego zbioru $(\varphi_{(q'')}/\varphi_{(q')'})$ do odpowiadających im pikseli w zbiorze $(\varphi_p/\varphi_{p'})$ oznaczonych kolorem ciemnozielonym. Zapisując matematycznie:

$$\varphi_{q''} = \arg \min_{q \in \phi} d(\varphi_{p'}, \varphi_{q'}) \quad (4.8)$$

Dystans pomiędzy dwoma skrawkami $d(\varphi_{p'}, \varphi_{q'})$ definiowany jest jako suma kwadratów różnic wszystkich odpowiadających sobie pikseli $\varphi_{p'}$ oraz $\varphi_{q'}$.

$$d(\varphi_{p'}, \varphi_{q'}) = \sum_{p \in \varphi_{p'}} \left(\varphi_{p'}(p) - \varphi_{q'}(p') \right)^2 \quad (4.9)$$

gdzie $\varphi(p)$ wartość jasności piksela p z badanego okna. W celu odrestaurowania obrazu powyższy schemat powtarzany jest aż do momentu wypełnienia całego obszaru Ω . Metoda wmalowywania pierwszy raz przedstawiona w [6] nie zakłada żadnego priorytetu ani kolejności wypełniania braków ustalonej na podstawie otoczenia maski. Przeprowadzana synteza obrazu pomimo wyznaczenia dla danego piksela trafnego dopasowania może prowadzić do akumulowanej przez algorytm niespójności obrazu.



Rysunek 4.2: Wpływ prostego schematu wypełniania braków na wynikową teksturę obrazu. Źródło [7].

W 2004 roku Criminisi w [7] korzysta z koncepcji próbkowania kawałków obrazu proponując nowe podejście do wyznaczania kolejności ich wmalowywania. Kolejność wypełniania wyznaczana jest na podstawie izolinii poziomu jasności obrazu. W każdej iteracji algorytmu dla pikseli sąsiadujących z maską wyznaczane są wartości priorytetu. Następnie wypełnienie odbywa się dla piksela o największym priorytecie. Dzięki nowemu podejściu izoliny będące granicami struktur obrazu są kontynuowane w miejsca odrestaurowywane.

Algorytm przedstawiony w [7] opiera się na powtarzaniu 3 głównych kroków w

każdej iteracji wykonywanej do momentu wypełnienia wszystkich pikseli ze zbioru Ω . Każda kolejna iteracja prowadzi do przesunięcia granicy maski w jej głąb. W trakcie próbkowania kawałków obrazu zbiór ϕ będący źródłem próbkowania nie zmienia się. Pierwsze dwa kroki w pojedynczej iteracji służą wyznaczeniu priorytetu $P(p)$ dla każdego piksela obrazu znajdującego się na granicy maski $\partial\Omega$, natomiast trzeci znalezieniu najlepszego dopasowania $\varphi_{q''}$. Priorytet $P(p)$ każdego piksela p znajdującego się na granicy maski wyznaczany jest z zależności:

$$P(p) = Ct(p) \cdot Dt(p) \quad (4.10)$$

Pierwszy krok w każdej wykonywanej iteracji związany jest z obliczeniem wartości funkcji $Ct(p)$ definiowanej:

$$Ct(p) = \frac{\sum_{q \in \varphi_p} C(q)}{|\varphi_p|} \quad (4.11)$$

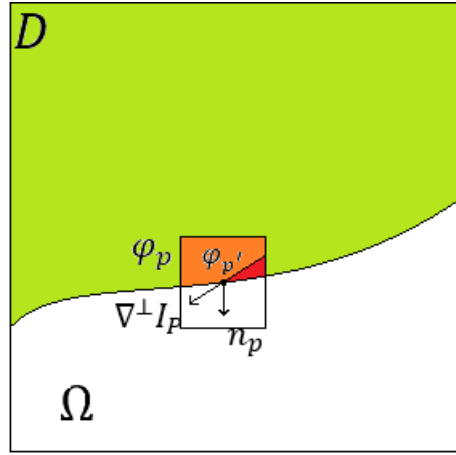
gdzie $C(q)$ - funkcja przypisująca wszystkim pikselom z obrazu poziom wiarygodności odwzorowania oryginalnego obrazu, $|\varphi_p|$ - liczba pikseli w analizowanym kawałku - rozmiar rozważanego okna.

Podczas inicjalizacji funkcja $C(q)$ dla każdego piksela stanowiącego oryginalny obraz przyjmuje wartość 1, natomiast dla punktów maski przyjmuje 0. Matematycznie $\forall_{q \in D} C(q) = 1$ i analogicznie $\forall_{q \in \Omega} C(q) = 0$. Wraz z wyznaczaniem kolejnych wartości pikseli z rejonu maski funkcja $C(q)$ jest aktualizowana, a zerowe wartości są nadpisywane. Funkcja $Ct(p)$ jest sumą wartości $C(p)$ wszystkich pikseli z analizowanego skrawka scentrowanego względem piksela p , które znajdują się w przestrzeni wyznaczonego już obrazu podzielonej przez ilość pikseli w danym skrawku. Funkcję interpretuje się jako czynnik wymuszający koncentryczne, czyli postępujące wypełnianie wzdłuż granicy maski nieznanych punktów zgodnie z pierwotnym założeniem w [6]. Wynika to z pierwszeństwa jakie wprowadza funkcja $C(p)$ przypisując w każdej iteracji coraz niższe wartości kolejnym wypełnianym pikselom. W ten sposób pierwszeństwo wypełniania uzyskują wycinki obrazu zawierające większą liczbę pikseli wypełnionych w poprzednich iteracjach, bądź zawierających oryginalne części obrazu. Mniejsza wartość $C(p)$ w każdym kroku interpretuje się jako mniejszą pewność co do przepisanych wartości koloru w obrazie. Drugi krok w

każdej wykonywanej iteracji związany jest z obliczenie wartości funkcji $D(p)$ definiowanej jako:

$$D(p) = \frac{|\nabla^\perp I_P \cdot n_p|}{\alpha} \quad (4.12)$$

gdzie człon $\nabla^\perp I_P$ podobnie jak w przypadku równania Naviera-Stokes'a odpowiada kierunkowi rozchodzenia się izolinii obrazu, natomiast n_p to wektor normalny (prostopadły) do granicy maski $\partial\Omega$ w punkcie piksela p . Zgodnie z Rysunek 4.3 funk-



Rysunek 4.3: Oznaczenie wektorów wykorzystywanych do liczenia wartości $D(p)$.

cja $D(p)$ w sposób liczbowy określa z jaką siłą izolinia obrazu uderza granicę maski. Ta zależna jest od gradientu jasności obrazu $\nabla^\perp I_P$ oraz orientacji względem gradientu wektora normalnego do granicy maski. Funkcja przyjmuje największą wartość w momencie, w którym wektor normalny oraz wektor izolinii są do siebie równoległe (zgodnie z iloczynem skalarnym wektorów $\cos 0^\circ = 1$), a człon $\nabla^\perp I_P$ ma największą wartość. W przeciwieństwie do definicji terminu $Ct(p)$ funkcję $D(p)$ można zinterpretować jako człon odpowiedzialny za przyznanie wyższego priorytetu pikselowi, stanowiącemu część struktury, która według psychologii widzenia powinna być kontynuowana w głąb odrestaurowywanego obszaru. Funkcja $D(p)$ wpływa na zmniejszenie pojawiającego się w metodzie zjawiska wprowadzania niespójności obrazu. Niech funkcja M w punktach oryginalnego obrazu D przyjmuje wartość 0 oraz w punktach maski w punktach Ω wartość 1, $I : D \rightarrow \{0, 1\}$. Wtedy korzystając z zależności (3.12) i (3.13) wektor normalny do granicy maski można wyznaczyć z

zależności:

$$n_p = \left[-n_x \frac{1}{L}, -n_y \frac{1}{L} \right] = \left[-\frac{M_{i+1,j} + M_{i-1,j}}{2h} \cdot \frac{1}{L}, -\frac{M_{i,j+1} + M_{i,j-1}}{2h} \cdot \frac{1}{L} \right] \quad (4.13)$$

$$L = \sqrt{n_x^2 + n_y^2} \quad (4.14)$$

Postać dyskretna równania (4.12) zgodnie z (3.14), (3.15), (4.13) i (4.14) przyjmuje postać:

$$D(p) = \left| u \cdot n_x \frac{1}{L} - v \cdot n_y \frac{1}{L} \right| \quad (4.15)$$

Znając wartości $Ct(p)$ oraz $D(p)$ dla każdego piksela z granicy maski można wyznaczyć wartość priorytetów $P(p)$. Na ich podstawie wybierane jest okno φ_p , a następnie najlepsze odwzorowanie φ_q . Ze względu na nieregularny kształt większości masek wyznaczenie jej granicy $\partial\Omega$ można wykonać poprzez operację erozji i dylacji na funkcji M . Innym sposobem może być zastosowanie dyskretnego splotu maski M z macierzą mL reprezentującą Laplasjan dla siatki 8-spójnej:

$$mL = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{vmatrix} \quad (4.16)$$

Podsumowując operację wmalowywania obrazu można przedstawić następującym algorytmem. Powtarzać aż do wypełnienia wszystkich pikseli z Ω :

1. Na podstawie (4.16) wyznaczyć granicę maski $\partial\Omega$
2. Na podstawie (4.10) wyznaczyć priorytety wszystkich pikseli z granicy maski $\partial\Omega$
3. Na podstawie (3.47) wyznaczyć najlepsze dopasowanie dla okna utworzonego na podstawie punktu 2
4. Zgodnie z Rysunek 4.1 przepisać wartości z okna najlepszego dopasowania w odrestaurowywany obszar, jednocześnie aktualizując wartości $C(p)$ zgodnie z zależnością:

$$C(p) = C(\dot{p}) \quad (4.17)$$

gdzie $p \in \varphi_p \cap \Omega$, a $C(p)$ to deprecjonowana wartość w każdej kolejnej iteracji.

Poprawa algorytmu przedstawiona w [7] prowadzi do polepszenia otrzymywanych wyników, lecz nie w optymalnym stopniu. W celu dalszej minimalizacji uzyskiwanych niespójności w obrazie i optymalizacji pod kątem szybkości działania algorytmów zostały wprowadzone kolejne modyfikacje: [8] [23] [1]. Jednym z istotnych ulepszeń algorytmu wmalowywania w oparciu o kopiowanie kawałków jest wprowadzenie zależności kolejności ich wstawiania w oparciu o główne struktury w obrazie. W niniejszej pracy zaimplementowano i sprawdzono odrestaurowywanie obrazu na podstawie zadawanych ręcznie przez użytkownika głównych linii zgodnie z [8]. Różnica pomiędzy zwykłym algorytmem przedstawionym w [7], a nowym rozwiązaniem polega na wstępnej definicji przez użytkownika grupy punktów znajdujących się w masce. Najczęściej są to krzywe, które zgodnie z psychologią widzenia powinny być kontynuowane w głąb maski tworząc odpowiednie połączenia między sobą. Krzywe charakteryzują się ograniczoną grupą pikseli mogących stanowić źródło uzupełniania maski. Ograniczając pole ϕ dla każdej kontynuowanej znacznie przyspiesza się krok iteracji związany z odnalezieniem najlepszej pary dopasowania $\varphi_{p'}$ oraz $\varphi_{q'}$. Schemat wmalowywania w oparciu o powyższy schemat przedstawia poniższy rysunek. W celu zaprojektowania w pełni automatycznego programu



Rysunek 4.4: Koncepcja wmalowywania w oparciu o główne struktury obrazu. Źródło [8]

do procesu wmalowywania opierającego się na algorytmie wprowadzonym przez Criminisi zaproponowano dodatkową analizę obrazu ograniczającą interakcję użytkownika wyłącznie do wprowadzenia maski. Autorzy w [1] proponują wykonać 2 dodatkowe kroki:

1. Wyznaczenie głównych struktur obrazu.
2. Rozwiązanie poniższych zależności:

$$D(i, j) \triangleq IP \cdot [\alpha \cdot D_1(f_C^i, f_C^j) + \beta \cdot D_2(f_T^i, f_T^j) + \gamma \cdot D_3(f_{Cur}^i, f_{Cur}^j)] \quad (4.18)$$

$$Paired(i, j) = \arg \min_{i, j \in \{1, \dots, N\}} D(i, j) \quad (4.19)$$

Analiza powyższych równań wykonana zostanie w szczegółowy sposób dla każdego z członów $\alpha \cdot D_1(f_C^i, f_C^j)$, $\beta \cdot D_2(f_T^i, f_T^j)$, oraz $\gamma \cdot D_3(f_{Cur}^i, f_{Cur}^j)$. Upřednim krokiem do powyższych równań według [1] jest wyznaczenie głównych struktur na podstawie obrazu wejściowego poddanemu wstępemu rozmyciu funkcją Gaussa:

$$G(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.20)$$

Operacji rozmycia dokonuje się poprzez operację splotu obrazu z odpowiednio wygenerowaną macierzą wyznaczoną na podstawie powyższego równania. Operacja rozmycia pozwala pozbyć się mniej znaczących krawędzi z obrazu zostawiając tylko najistotniejsze cechy, które powinny być kontynuowane w niewyznaczony obszar. Po operacji rozmycia autorzy w [1] proponują wykrycie głównych struktur na podstawie transformacji falkowej oraz odpowiednio wyznaczonych lokalnych maksimów dla funkcji powstałych na podstawie wspomnianej transformacji. Na podstawie przeprowadzonej operacji uzyskuje się wszystkie główne struktury obrazu. Zbiorem krawędzi poddanych do analizy są ostatecznie struktury, dla których dochodzi do przecięcia ze zdefiniowaną maską Ω zgodnie z poniższym rysunkiem (jako maska traktowany jest samochód). Dla każdej linii stykającej się z obszarem Ω wyznaczony zostaje punkt środkowy. Względem środka tego punktu tworzone jest okno f^i o upřednio zdefiniowanym obszarze. Pierwszym członem w równaniu [4.19][4.19] wyznaczonym na podstawie utworzonego okna f^i jest $D_1(f_C^i, f_C^j)$. Symbole f_C^i oraz f_C^j stanowią posegmentowane fragmenty obrazu względem upřednio zdefiniowanej liczby głównych, dominujących kolorów w obrazie M . Dla każdego wycinka f_C^i można zdefiniować zbiór punktów (c_k, p_k) , gdzie c_k odpowiada danemu kolorowi, p_k jego procentowemu udziałowi we fragmencie f^i , gdzie $k = 1, \dots, M$.

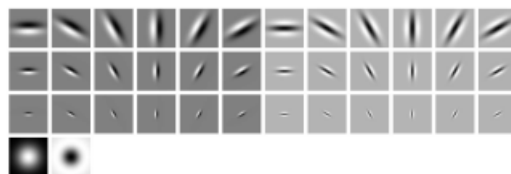


Rysunek 4.5: Koncepcja wmalowywania w oparciu o główne struktury obrazu. źródło [8].

Dysponując zbiorami punktów dla okien f_C^i, f_C^j można liczbowo przedstawić ich poziom podobieństwa obliczając:

$$D_1(f_C^i, f_C^j) = IOCCD((c_k, p_k)_i, (c_k, p_k)_j) \quad (4.21)$$

Dokładna analiza oraz sposób wyznaczania podobieństwa okien f^i przedstawiony jest w [24]. Drugim członem w równaniu [4.19][4.19] jest $D_2(f_T^i, f_T^j)$ określający stopień podobieństwa tekstur znajdujących się w utworzonych oknach f . Dla każdego obszaru f^i z wyznaczonego zbioru okien wyznaczana jest macierz $MR8(f^i)$. Macierz uzyskiwana jest na podstawie odpowiedzi fragmentu obrazu na grupę 38 filtrów przedstawionych na poniższym rysunku. Przez odpowiedź obrazu na filtr



Rysunek 4.6: Bank 38 filtrów grupy MR8. źródło [25].

należy rozumieć operację splotu z macierzą odpowiadającą definicji konkretnego filtru. W grupie filtrów $MR8$ branych pod uwagę znajduje się filtr Gaussowski, oraz

filtr *LOG* będący laplasjanem funkcji gaussowskiej. Na pozostałe 36 filtrów składają się dwie grupy filtrów utworzonych na podstawie filtru krawędziowego oraz filtru prętowego. Każdy z nich przedstawiony jest w sześciu orientacjach oraz 3 skalach zgodnie z rysunkiem Rysunek 4.6. Jako maksymalną odpowiedź na bazę trzydziestu ośmiu filtrów rozumie się zależność:

$$I_{fmax}(p) = \max(I_{f1}(p), I_{f2}(p), \dots, I_{f38}(p)) \quad (4.22)$$

gdzie p - piksel obrazu poddanego filtracji, $p \in I$, I_{fi} - odpowiedź obrazu I na i -ty filtr z bazy filtrów. Podobieństwo wyznaczonych okien odpowiadających głównym strukturom w obrazie można wyrazić zależnością:

$$D_2(f_T^i, f_T^j)_i = \frac{|MR8(f^i) - MR8(f^j)|_F}{250} = \frac{|f_T^i - f_T^j|_F}{250} \quad (4.23)$$

W powyższej zależności norma oznaczona literą F odpowiada normie Frobenius'a liczonej następująco:

$$|I| = \sqrt{\sum_{p \in I} |I(p)|^2} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (f_T^i(i, j) - f_T^j(i, j))^2} \quad (4.24)$$

Dokładny algorytm klasyfikacji tekstur przedstawiony został przez Manika Varma'e i Adrew Zisserman'a w [25]. Ostatnim członem w równaniu (4.18) jest człon $D_3(f_{cur}^i, f_{cur}^j)$ odpowiadający liczbowej reprezentacji podobieństwa krzywizn ze zbioru głównych struktur. Autorzy w [1] nie podają sposobu wyznaczenia normy:

$$D_3(f_{cur}^i, f_{cur}^j) = |curvature_i - curvature_j| \quad (4.25)$$

Proponują natomiast każdą z wyznaczonych głównych struktur przedstawić w postaci równania wielomianu drugiego stopnia. Aproksymacji współczynników wielomianu można dokonać na podstawie rozwiązania poniższego równania macierzowego:

$$\begin{vmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{vmatrix} \cdot \begin{vmatrix} a_0 \\ a_1 \\ a_2 \end{vmatrix} = \begin{vmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{vmatrix} \quad (4.26)$$

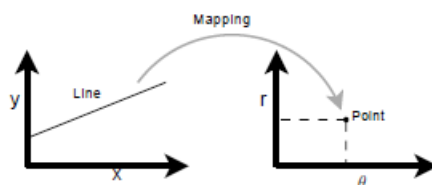
gdzie n określa liczbę zastosowanych par (x_i, y_i) , a wielomian przedstawia się w postaci:

$$curvature_i = f_i(x) = a_{0_i} + a_{1_i}x + a_{2_i}x^2 \quad (4.27)$$

Według wyznaczonych równań, pary głównych struktur uzyskane w dalszych krokach algorytmu, propagowane są w głąb przestrzeni Ω , aż do momentu ich przecięcia. Ostatnią ważną zmienną w równaniu (4.18) jest IP . Jest to zmienna, która przyjmuje wartość zerową, gdy badana para głównych struktur nie jest ze sobą równoległa, bądź jeden w przeciwnym wypadku. Aby uzyskać równania prostych opisujące wyznaczone krzywe można dokonać aproksymacji na podstawie transformacji Hough'a. Transformacja bazuje na wygodnej postaci równania prostej:

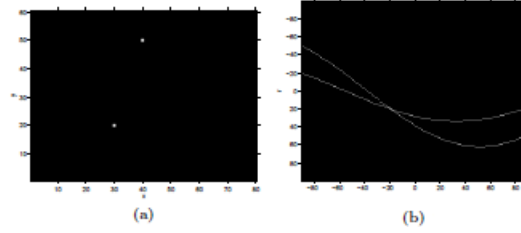
$$y = -\frac{\cos\theta}{\sin\theta}x + \frac{r}{\sin\theta} \quad (4.28)$$

Posługiwanie się parametrami θ oraz r pozwala uniknąć niejednoznaczności, w przypadku opisu pionowych linii oraz umożliwia przedstawienie prostej w postaci jednego punktu w przestrzeni Hough'a. Mapowanie z przestrzeni D można przedstawić następująco: Posługując się parametrem θ określającym kąt pomiędzy wy-



Rysunek 4.7: Odwzorowanie prostej w przestrzeni Hough'a

znaczoną prostą a osią x łatwo określić czy badane proste są do siebie równoległe. Aby wyznaczyć parametry prostej z równania (4.28) zgodnie z [26] należy każdemu punktowi tworzącemu prostą przypisać grupę punktów z przestrzeni Hough'a dla których wszystkie możliwe proste utworzone na ich podstawie zawierałyby badany punkt. Ostatecznie parametry prostej pochodzą z punktu, dla którego w przestrzeni Hough'a występuje największa gęstość powtarzających się punktów. Transformację przedstawia poniższy rysunek: Dokładną analizę transformacji można znaleźć



Rysunek 4.8: Transformacja dwóch punktów do dwóch krzywych w przestrzeni Hough'a. Źródło [26].

w [26]. Pojawiające się współczynniki α, β, γ w równaniu (4.18) stanowią uprzednio definiowane wartości przypisujące procentowy wpływ poszczególnych członów na ostateczną wartość dopasowania $D(i, j)$. Według [1] najważniejszym dopasowaniem jest dopasowanie kolorów, następnie tekstur i krzywizn. Stąd należy przyjmować $\alpha > \beta > \gamma$ gdzie suma współczynników wynosi 1. W pracy zbadano powyższe rozwiązanie w ograniczeniu do wykorzystania członów $D_1(f_C^i, f_C^j)$ oraz IP . Człon $D_2(f_T^i, f_T^j)$ jest istotny w przypadku obrazów monochromatycznych o różnej teksturze. W przypadku obrazów kolorowych w większości przypadków linie granicy głównych struktur tworzone są poprzez gradienty kolorów. Zaletą metody jest stosowanie podzbiorów stanowiących źródło kopiowanych skrawków w nieznaną część obrazu. W uproszczeniu można to przedstawić zgodnie z rysunkiem: Pomi-



Rysunek 4.9: Grupowanie podzbiorów stanowiących źródło skrawków wykorzystywanych do syntezy podzbiorów nieznanego obszaru Ω . źródło [8].

mo automatycznej metody wykrywania i kontynuowania głównych struktur algorytm wymaga wielu operacji wykonywanych zdalnie przez użytkownika, takich jak stopień rozmycia wymagany do wprowadzenia w obrazie, wielkość kopiowanych

skrawków ψ , czy też okien stanowiących podstawę do parowania danych struktur f . W przeciwieństwie do metod bazujących na rozwiązywaniu równań różniczkowych oraz metod bazujących na procesie dyfuzji proces wmalowywania opisany w aktualnym rozdziale pozbawiony jest wady w postaci rozmycia w odrestaurowanym obszarze. Jednym z problemów metody jest konieczność stosowania metody korekcji fotometrycznej zgodnie z [8], wygładzającej lokalne granice obrazu powstałe w wyniku syntezy skrawków obrazu ψ w wyznaczanym obszarze Ω .

Rozdział 5

Wmalowywanie struktury i tekstury obrazu.

Metoda jednoczesnego wmalowywania struktur i tekstur stanowi połączenie wcześniej opisanych w pracy metod opartych o równania różniczkowe wyższego rzędu z metodami bazującymi na przykładowych źródłach kawałków obrazów stanowiących podstawę do wypełniania brakujących regionów. Pierwszy raz przedstawiona została przez Bertalmio i innych w 2003 roku w [9], a bardziej zaawansowany algorytm pojawia się w 2004 roku w [26]. Metoda w [9] opiera się na wmalowywaniu struktury w oparciu o algorytm przedstawiony w [2] przy jednoczesnej propozycji prostej syntezy tekstury.

Niech obraz $I : R^2 \rightarrow R$, $I \in L^2(R^2)$. Proces wypełniania brakującego regionu Ω poprzedzony jest dekompozycją obrazu:

$$I(i, j) = u(i, j) + v(i, j) \quad (5.1)$$

uzyskaną na podstawie rozwiązania problemu minimalizacji energii:

$$\min_u \left\{ E(u) = \int |\nabla u| + \lambda \|v\|_{L^2}^2 \right\} \quad (5.2)$$

gdzie: $\int |\nabla u|$ stanowi funkcjonal z przestrzeni $BV(R^2)$ pozwalający usunąć z oryginalnego obrazu szum i powtarzające się w małej skali detale, przy jednoczesnym

zachowaniu głównych cech oraz ostrych krawędzi, $\|v\|_{L^2}^2$ człon odpowiadający dokładności odwzorowania, λ stanowi uprzednio definiowany parametr. Dla odpowiednio małych wartości λ problem minimalizacji prowadzi do uzyskania jednorodnego komponentu u reprezentującego odfiltrowany obraz oraz komponentu v reprezentującego wahania w postaci szumu oraz tekstur. Zgodnie z dowodem Mayera w przypadku gdy pożądanym jest wspomniany kształt funkcji v , można udowodnić jej przynależność do przestrzeni Banacha oznaczanej G . W związku z tym (pomijając wyprowadzenia) problem minimalizacji można przedstawić w ostatecznej postaci minimalizacji energii:

$$\min_{u, g_1, g_2} \left\{ E = \int |\nabla u| + \lambda \int |I - u - \partial_x g_1 - \partial_y g_2|^2 dx dy + \mu \left[\int \sqrt{g_1^2 + g_2^2} dx dy \right]^{\frac{1}{p}} \right\} \quad (5.3)$$

gdzie:

$$v = \operatorname{div} \left(\vec{g} \right), \quad \vec{g} = (g_1, g_2), \quad g_1 = -\frac{1}{2\lambda} \frac{u_x}{|\nabla u|}, \quad g_2 = -\frac{1}{2\lambda} \frac{u_y}{|\nabla u|} \quad (5.4)$$

oraz $p \rightarrow \infty$. Stosując podstawowe równanie rachunku wariacyjnego postaci równania Eulera-Lagrange'a uzyskuje się odpowiadające mu postacie funkcji wykorzystywane do minimalizacji energii:

$$u = I - u - \partial_x g_1 - \partial_y g_2 + \frac{1}{2\lambda} \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad (5.5)$$

$$\mu H(g_1, g_2) g_1 = 2\lambda \left[\frac{\partial}{\partial x} (u - I) + \partial_{xx}^2 g_1 + \partial_{xy}^2 g_2 \right] \quad (5.6)$$

$$\mu H(g_1, g_2) g_2 = 2\lambda \left[\frac{\partial}{\partial y} (u - I) + \partial_{xy}^2 g_1 + \partial_{yy}^2 g_2 \right] \quad (5.7)$$

gdzie μ stanowi uprzednio definiowany parametr dyskretyzacji, $H(g_1, g_2)$ opisuje się wzorem:

$$H(g_1, g_2) = \left\| \sqrt{g_1^2 + g_2^2} \right\|_p^{1-p} \left(\sqrt{g_1^2 + g_2^2} \right)^{p-2} \quad (5.8)$$

Zgodnie z [27] w celach dyskretyzacji parametr p przyjęto równy 1, ze względu na brak znaczących różnic wyniku w przypadku stosowania większych wartości. Wtedy:

$$H(g_1, g_2) = \frac{1}{\sqrt{g_1^2 + g_2^2}} \quad (5.9)$$

Formalnie, jednym z równań Eulera-Lagrange'a uzyskiwanym podczas minimalizacji wspomnianej energii E jest:

$$u = I + \frac{1}{2\lambda} \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad (5.10)$$

Stąd można wywnioskować:

$$v = -\frac{1}{2\lambda} \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad (5.11)$$

Korzystając z zależności przedstawionych w (5.4) łatwo zauważyć $g_1^2 + g_2^2 = \frac{1}{2\lambda}$. Stąd:

$$H(g_1, g_2) = \sqrt{2\lambda} \quad (5.12)$$

$$u = I - u - \partial_x g_1 - \partial_y g_2 + \frac{1}{2\lambda} \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) \quad (5.13)$$

$$\mu\sqrt{2\lambda}g_1 = 2\lambda \left[\frac{\partial}{\partial x} (u - I) + \partial_{xx}^2 g_1 + \partial_{xy}^2 g_2 \right] \quad (5.14)$$

$$\mu\sqrt{2\lambda}g_2 = 2\lambda \left[\frac{\partial}{\partial y} (u - I) + \partial_{xy}^2 g_1 + \partial_{yy}^2 g_2 \right] \quad (5.15)$$

Jako wartości początkowe iteracyjnego algorytmu można przyjąć następujące przypuszczenie:

$$u^0 = f, \quad g_1 = -\frac{1}{2\lambda} \frac{f_x}{|\nabla f|}, \quad g_2 = -\frac{1}{2\lambda} \frac{f_y}{|\nabla f|} \quad (5.16)$$

Następnie stosując zależności wyznaczania pochodnych (3.12), (3.13), (3.25), (3.26) oraz poniższą metodę aproksymacji pochodnej drugiego stopnia:

$$\begin{aligned} \partial_{xy} g_1(i, j) \cong \frac{1}{2h^2} & \left[2g_1(i, j) + g_1(i-1, j-1) + g_1(i+1, j+1) - g_1(i, j-1) \right. \\ & \left. - g_1(i-1, j) - g_1(i+1, j) - g_1(i, j+1) \right] \end{aligned} \quad (5.17)$$

równania (5.13), (5.14) i (5.15) można przedstawić w następującej dyskretnej postaci:

$$\begin{aligned}
 u^{n+1}(i, j) = & \left[\frac{1}{1 + \left(\frac{1}{2\lambda h^2}\right) \cdot (c_1(i, j) + c_2(i, j) + c_3(i, j) + c_4(i, j))} \right] \\
 & \cdot \left[I(i, j) - \left(\frac{g_1^n(i+1, j) \cdot g_1^n(i-1, j)}{2h} \right) - \left(\frac{g_2^n(i, j+1) - g_2^n(i, j-1)}{2h} \right) \right. \\
 & + \left(\frac{1}{2\lambda h^2} \right) \cdot \left[c_1(i, j) \cdot u^n(i+1, j) + c_2(i, j) \cdot u^n(i-1, j) \right. \\
 & \left. \left. + c_3(i, j) \cdot \lambda \cdot u^n(i, j+1) + c_4(i, j) \cdot u^n(i, j-1) \right] \right]
 \end{aligned} \tag{5.18}$$

$$\begin{aligned}
 g_1^{n+1}(i, j) = & \left[\frac{2\lambda}{\mu\sqrt{2\lambda} + 4\frac{\lambda}{2h}} \right] \cdot \left[\frac{u^n(i+1, j) - u^n(i-1, j)}{2h} - \frac{I(i+1, j) - I(i-1, j)}{2h} \right. \\
 & + \frac{g_1^n(i+1, j) + g_1^n(i-1, j)}{h^2} \frac{1}{2h^2} \cdot \left[2g_2^n(i, j) + g_2^n(i-1, j-1) \right. \\
 & + g_2^n(i+1, j+1)g_2^n(i, j-1) - g_2^n(i-1, j) + g_2^n(i+1, j) \\
 & \left. \left. - g_2^n(i, j+1) \right] \right]
 \end{aligned} \tag{5.19}$$

$$\begin{aligned}
 g_2^{n+1}(i, j) = & \left[\frac{2\lambda}{\mu\sqrt{2\lambda} + 4\frac{\lambda}{2h}} \right] \cdot \left[\frac{u^n(i, j+1) - u^n(i, j-1)}{2h} - \frac{I(i, j+1) - I(i, j-1)}{2h} \right. \\
 & + \frac{g_2^n(i, j+1) + g_2^n(i, j-1)}{h^2} + \frac{1}{2h^2} \cdot \left[2g_1^n(i, j) + g_1^n(i-1, j-1) + g_1^n(i+1, j+1) \right. \\
 & \left. \left. - g_1^n(i, j-1) - g_1^n(i-1, j) + g_1^n(i+1, j) - g_1^n(i, j+1) \right] \right]
 \end{aligned} \tag{5.20}$$

gdzie:

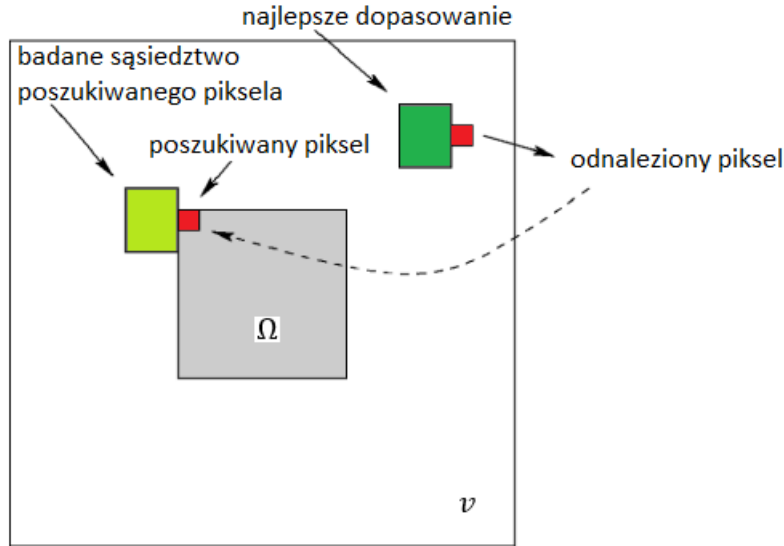
$$c_1^n(i, j) = \frac{1}{\sqrt{\left(\frac{u^n(i+1, j) - u^n(i, j)}{h}\right)^2 + \left(\frac{u^n(i, j+1) - u^n(i, j-1)}{2h}\right)^2} + \varepsilon} \tag{5.21}$$

$$c_2^n(i, j) = \frac{1}{\sqrt{\left(\frac{u^n(i, j) - u^n(i-1, j)}{h}\right)^2 + \left(\frac{u^n(i-1, j+1) - u^n(i-1, j-1)}{2h}\right)^2} + \varepsilon} \tag{5.22}$$

$$c_3^n(i, j) = \frac{1}{\sqrt{\left(\frac{u^n(i+1, j) - u^n(i-1, j)}{2h}\right)^2 + \left(\frac{u^n(i, j+1) - u^n(i, j)}{h}\right)^2 + \varepsilon}} \quad (5.23)$$

$$c_4^n(i, j) = \frac{1}{\sqrt{\left(\frac{u^n(i+1, j-1) - u^n(i-1, j-1)}{2h}\right)^2 + \left(\frac{u^n(i, j) - u^n(i, j-1)}{h}\right)^2 + \varepsilon}} \quad (5.24)$$

W celu uniknięcia problemu dzielenia przez zero w powyższej dyskretyzacji wprowadzono parametr $\varepsilon \rightarrow 0$. Po przeprowadzeniu powyższej segmentacji obrazu dla uzyskanych członów u i v można zastosować różne algorytmy wmalowywania braków. Zgodnie z ich charakterem najodpowiedniejszymi metodami wmalowywania struktur są metody oparte na równaniach różniczkowych, natomiast uzyskanie brakującej tekstury najlepiej wykonać metodą kopii kawałków obrazów z najlepszym dopasowaniem. W [9] autorzy proponują następujący algorytm syntezy brakującej tekstury Ω : Zgodnie z Rysunek 5.1 dla każdego piksela znajdującego się na granicy



Rysunek 5.1: Prosty algorytm wmalowywania tekstury obrazu.

Ω tworzone jest okno stanowiące szablon wykorzystywany do odnalezienia najlepszego dopasowania z pośród całego obrazu v . W przypadku odnalezienia takiego dopasowania uzyskany piksel kopiowany jest w miejsce poszukiwanego piksela. Algorytm można wykonywać pobierając piksele z granicy w kolejności zgodnej ze

wskazówkami zegara, postępując w głąb obszaru Ω , aż do wypełnienia wszystkich pikseli. Odnalezienie okna najlepszego dopasowania można wykonać analogicznie do metody wykorzystanej w publikacji Criminisi’ego, w oparciu o zależności (4.8) oraz (3.47).

W przypadku obrazów kolorowych pierwszym krokiem wykonania operacji wmalowywania jest transformacja do współrzędnych sferycznych zgodnie z zależnościami (3.38), (3.39) i [?] ze względu na ich lepsze właściwości. Następnie każda z trzech uzyskanych warstw jest rozbijana na własną teksturę i strukturę. W przypadku algorytmu dokonującego syntezy tekstury oddzielnie dla każdej z 3-ech segmentowanych warstw uzyskiwane wyniki charakteryzują się pojawiającymi się niespójnościami kolorów. Stąd w przypadku odrestaurowywania uzyskanych części v , zlokalizowanie piksela będącego najlepszym dopasowaniem aktualnie badanego, nieznanego piksela, odbywa się na podstawie pierwszej warstwy, a wszystkie 3 uzupełniane są analogicznie do obliczeń uzyskanych na podstawie pierwszej warstwy. Kroki algorytmu można przedstawić następująco:

1. Przekształcenie warstw R, G, B obrazu na współrzędne sferyczne I_1, I_2, I_3
2. Wyznaczenie $u_1, v_1, u_2, v_2, u_3, v_3$ na podstawie I_1, I_2, I_3
3. Wmalowanie nieznanymi obszarów Ω w warstwach u_1, u_2, u_3 metodą adaptacji równania Navier-Stokes’a
4. Wmalowanie obszaru Ω w v_1, v_2, v_3 na podstawie prostego algorytmu syntezy tekstury najlepszymi dopasowaniami uzyskanymi na podstawie warstwy v_1

Rozdział 6

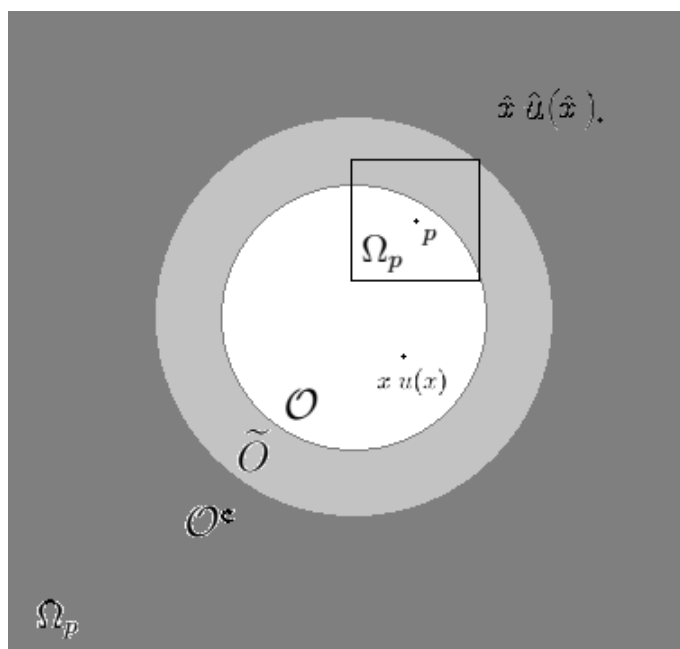
Wariacyjny, nielokalny model wmalowywania

Ostatnią z rozważanych metod będzie metoda zaproponowana przez autorów Fedorov, Facciolo i Arias w [10]. Autorzy w publikacji proponują optymalizację energii $\mathcal{E}(\varphi(x), u(\mathcal{O}))$ zależnej od dwóch niewiadomych:

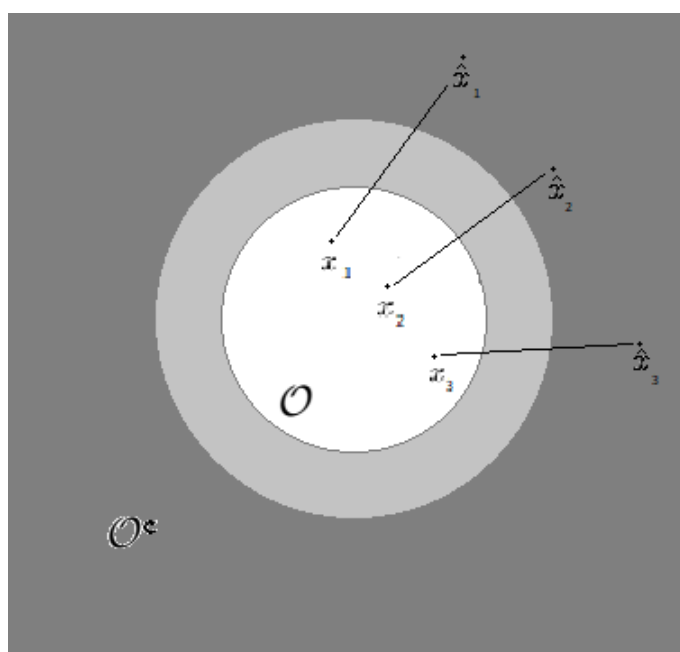
- nieznanego obszaru obrazu $u(\mathcal{O})$
- mapy podobieństwa będącej funkcją przypisującą każdemu punktowi z nieznanego obszaru jego odpowiednika z obszaru poza maską $\varphi(\hat{x})$

Zgodnie z Rysunek 6.1 niech obraz stanowi przekształcenie $u : \Omega \rightarrow R$, \mathcal{O} oznacza obszar do uzupełnienia, \mathcal{O}^c oryginalną część obrazu nie ulegającą modyfikacji, Ω_p podzbiór pikseli scentrowanych względem piksela $p \in \Omega$, gdzie $\Omega = \mathcal{O}^c + \mathcal{O}$, $\tilde{\mathcal{O}} := \mathcal{O} + \Omega_{ps}$, gdzie Ω_{ps} oznacza sumę wszystkich zbiorów punktów $x \in \mathcal{O}$. Podobnie jak w [10] niech przekształcenie $u(x)$ jest zarezerwowane dla przestrzeni maski, a $\hat{u}(\hat{x})$ odpowiada przekształceniu w przestrzeni stałej obrazu $\tilde{\mathcal{O}}$ ($\hat{x} \in \mathcal{O}$). Mapowanie funkcji podobieństwa $\varphi(x)$ przedstawia Rysunek 6.2. Energię poddawaną minimalizacji przedstawia się zależnością:

$$\mathcal{E}(u, \varphi) = \int_{\tilde{\mathcal{O}}} \int_{\Omega_p} |u(x+h) - \hat{u}(\varphi(x)+h)|^2 dh dx \quad (6.1)$$



Rysunek 6.1: Określenie charakterystycznych obszarów obrazu.



Rysunek 6.2: Funkcja podobieństwa $\varphi(\hat{x}) : x_1 = \varphi(\hat{x}_1), x_2 = \varphi(\hat{x}_2), x_3 = \varphi(\hat{x}_3)$.

Funkcja jest niewypukła i pozwala obliczyć jedynie lokalne minima. W pracy wykorzystano iteracyjny algorytm przemiennej minimalizacji polegający na wyznaczaniu w każdym kroku $u(x)$ względem stałej wartości $\varphi(x)$, a następnie $\varphi(x)$ względem

stałej wartości $u(x)$.

6.1 Wyznaczenia mapy podobieństwa.

Proces minimalizacji funkcji względem stałej zmiennej $u(x)$ można opisać zależnością:

$$\arg \min_t \mathcal{E}(p_u(x) - p_{\hat{u}}(x+t)), \text{ gdzie } \forall_t : x+t \in \mathcal{O}^c \quad (6.2)$$

Zmienna t w minimalizacji stanowi zbiór wartości, który po dodaniu do współrzędnej x mapuje do dowolnego położenia w \mathcal{O}^c . Wynikiem minimalizacji jest φ , która stanowi funkcję wyznaczającą najbliższego sąsiada punktu x z przestrzeni \mathcal{O}^c , $\Theta(x) : \mathcal{O} \rightarrow \mathcal{O}^c$. Warto zaznaczyć, iż najbliższego sąsiada punktu x nie rozumie się znajdującego się najbliżej, a tego dla którego zgodnie z przyjętą normą uzyskuje się najniższą wartość z pośród wszystkich możliwych:

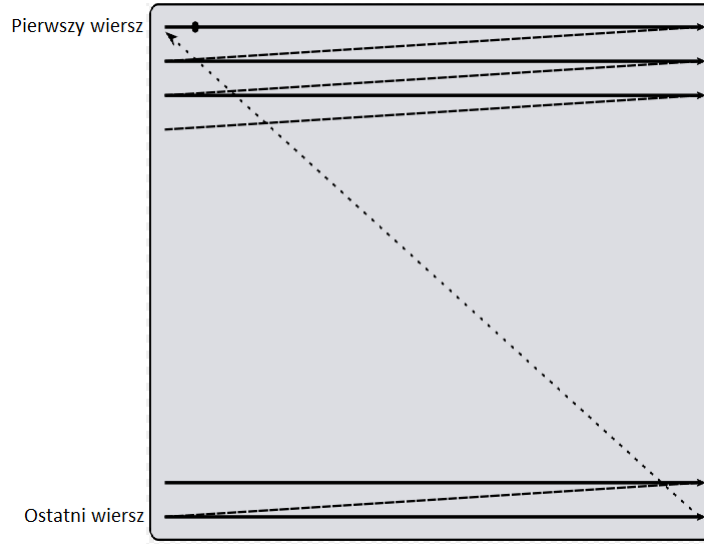
$$\|p_u(x) - p_{\hat{u}}(\hat{x})\| \quad (6.3)$$

Norma może być wyznaczana zgodnie z (3.46) bądź (3.47). Autorzy w [4] proponują dodatkowe definicje norm, które zostaną opisane w dalszej części pracy. Najprostszy algorytm wyznaczania najbliższego sąsiada może osiągnąć złożoność do $O(N^3)$ i stanowić duże obciążenie obliczeniowe. Autorzy w [10] proponują zastosowanie algorytmu PatchMatch dokładnie opisanego w [28]. Algorytm oparty jest na 2 krokach (punkt x oznaczany będzie zamiennie ze współrzędną (i, j) w obrazie):

1. Losowej inicjalizacji pikseli
2. Ograniczonej uprzednio definiowaną liczbą iteracji powtarzanej czynności:
 - a) propagacja: w przypadku parzystej iteracji sprawdzenie wszystkich pikseli x z maski w kierunku rastrowym przedstawionym na Rysunek 6.3, czy wartość $\|p_u(i, j) - p_{\hat{u}}(\varphi(i+1, j))\|$, bądź $\|p_u(i, j) - p_{\hat{u}}(\varphi(i, j+1))\|$ nie jest mniejsza od obecnie wyznaczonej i w takim przypadku nadpisaniu najbliższego sąsiada $\varphi(i, j) = \varphi(i+1, j)$, lub $\varphi(i, j) = \varphi(i, j+1)$, w przypadku nieparzystej

iteracji sprawdzenie wszystkich pikseli x w kierunku odwrotnym do rastrowego przedstawionym na Rysunek 6.3, czy wartość $\|p_u(i, j) - p_{\hat{u}}(\varphi(i-1, j))\|$ bądź $\|p_u(i, j) - p_{\hat{u}}(\varphi(i, j-1))\|$ nie jest mniejsza od obecnie wyznaczonej i w takim przypadku nadpisaniu najbliższego sąsiada $\varphi(i, j) = \varphi(i-1, j)$, lub $\varphi(i, j) = \varphi(i, j-1)$

- b) poszukiwanie w oknach: dla każdego punktu (i, j) sprawdzane jest, czy w stopniowo pomniejszanych oknach centrowanych względem punktu $\varphi(i, j)$ istnieje losowy punkt (i', j') , dla którego zachodzi $\|p_u(i, j) - p_{\hat{u}}(i', j')\| < \|p_u(i, j) - p_{\hat{u}}(\varphi(i, j))\|$, który w następstwie jest podmieniany.



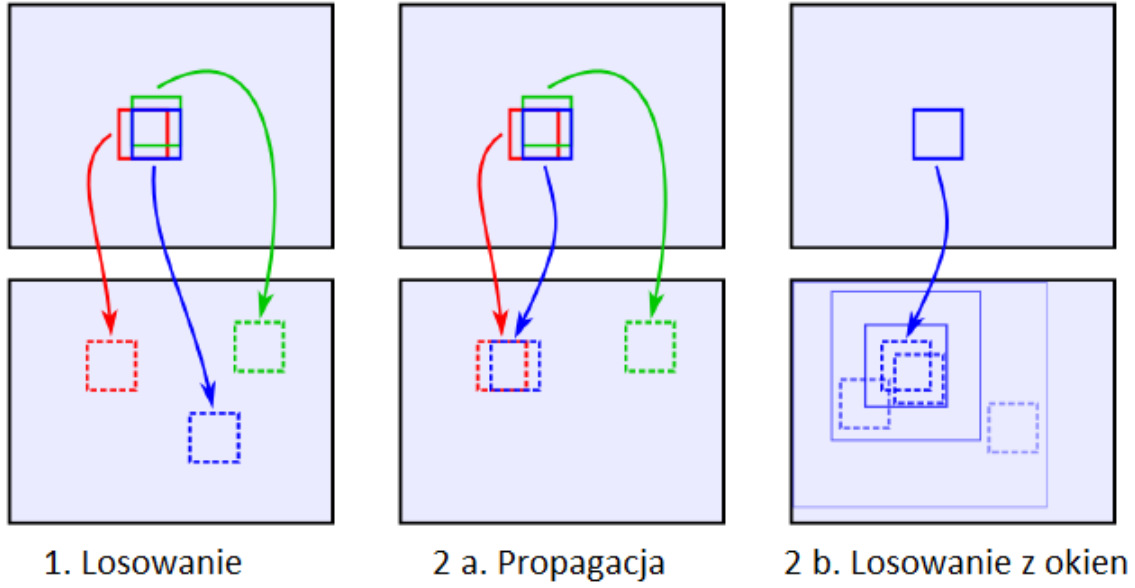
Rysunek 6.3: Funkcja podobieństwa $\varphi(\hat{x}) : x_1 = \varphi(\hat{x}_1), x_2 = \varphi(\hat{x}_2), x_3 = \varphi(\hat{x}_3)$.

Kroki algorytmu PatchMatch zostały przedstawione graficznie na Rysunek 6.4.

6.2 Wyznaczenie wartości obrazu.

Niech funkcja $e : \Omega_p \rightarrow \mathbb{R}^+$ stanowi sumę ważonych błędów pomiędzy dwoma skrawkami obrazu scentrowanymi względem punktów x i \hat{x} .

$$\begin{aligned} e(x, \hat{x}) &= g * \|p_u(x) - p_{\hat{u}}(\hat{u})\| = g * \|u(x + \cdot) - u(\hat{x} + \cdot)\| \\ &= \int_{\Omega_p} g(h) \|u(x + h) - \hat{u}(\hat{x} + h)\| dh \end{aligned} \quad (6.4)$$



Rysunek 6.4: Algorytm PatchMatch. Źródło [10].

Autorzy w [10] proponują minimalizację następującej energii będącej akumulacją błędów pikseli:

$$E(u) = \int_{\tilde{O}} \int_{\Omega_p} g(h) e(u(x+h) - \hat{u}(\varphi(x)+h)) dh dx \quad (6.5)$$

Niech $z := x + h$ i $\hat{z} := \hat{x} + h$. Wprowadzając zależność:

$$m(z, \hat{z}) = \int_{\Omega_p} g(h) \chi_{\tilde{O}}(z-h) \delta_{\varphi(z-h)}(\hat{z}-h) dh \quad (6.6)$$

gdzie:

1. $\chi(p)$ podobnie jak w modelu *NLCTV* przyjmuje wartość 1 poza maską i wartość 0 w masce.
2. $\delta(p)$ podobnie jak w metodzie Criminisi określa pewność co do danego piksela wewnątrz maski i jest określone następującą zależnością:

$$\delta(x) = (1 - c_0) \exp\left(\frac{-d(x, \partial\mathcal{O})}{t_c}\right) + c_0 \quad (6.7)$$

wartość t_c określa szybkość opadania funkcji eksponentialnej, natomiast c_0 asymptotę, do której to funkcja zmierza. Odległość punktu od granicy maski określona

jest zależnością $\frac{-d(x, \partial \mathcal{O})}{t_c}$. W zależności od zastosowanej normy (6.3) rozwiązanie problemu minimalizacji energii uzyskuje różne postacie. Autorzy w [10] przedstawiają rozwiązanie dla następujących norm:

1. Nielokalnej uśrednionej:

$$\|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,2}^2 = g * |u(x + \cdot) - \hat{u}(\hat{x} + \cdot)|^2 \quad (6.8)$$

2. Nielokalnej medianowej:

$$\|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,1} = g * |u(x + \cdot) - \hat{u}(\hat{x} + \cdot)| \quad (6.9)$$

3. Nielokalnej Poissona:

$$\begin{aligned} \|p_u(x) - p_{\hat{u}}(\hat{x})\|_P &= \|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,2}^2 + \|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,2,\nabla}^2 \\ &= g * (\lambda |u(x + \cdot) - \hat{u}(\hat{x} + \cdot)| \\ &\quad + (1 - \lambda) |\nabla u(x + \cdot) - \nabla \hat{u}(\hat{x} + \cdot)|) \end{aligned} \quad (6.10)$$

Suma różnic pikseli w przypadku normy Poissona dodatkowo wymaga wyznaczenia dywergencji obrazu. Współczynnik λ określa w jakim stopniu poszczególny z członów będzie miał wpływ na ostatecznie uzyskiwany wynik. Dla norm uzyskuje się odpowiednio rozwiązania w postaci:

1. równanie:

$$u(z) = \frac{1}{k(z)} \int_{\tilde{\mathcal{O}}^c} m(z, \hat{z}) \hat{u}(\hat{z}) d\hat{z} \quad (6.11)$$

gdzie

$$k(z) := \int_{\tilde{\mathcal{O}}^c} m(z, \hat{z}) d\hat{z} \quad (6.12)$$

2. równanie Eulera:

$$[\delta_u \mathcal{E}(u)](z) = \int_{\mathcal{O}^c} \text{sign}[u(z) - \hat{u}(\hat{z})] m(z, \hat{z}) d\hat{z} \ni \mathcal{O} \quad (6.13)$$

3. równania Eulera rozwiązane względem zmiennej u :

$$\begin{cases} \nabla \cdot [k(z)\nabla u(z)] - \frac{\lambda}{1-\lambda}k(z)u(z) = \\ \nabla \cdot [k(z)\nabla v(z)] - \frac{\lambda}{1-\lambda}k(z)f(z), & z \in \mathcal{O} \\ u(z) = \hat{u}(z), & z \in \partial\mathcal{O} \setminus \partial\Omega \\ \nabla u(z) \cdot n_\Omega(z) = 0, & z \in \partial\mathcal{O} \cap \partial\Omega \end{cases} \quad (6.14)$$

gdzie:

$$v(z) := \frac{1}{k(z)} \int_{\mathcal{O}^c} m(z, \hat{z}) \nabla \hat{u}(\hat{z}) d\hat{z} \quad (6.15)$$

$$f(z) := \frac{1}{k(z)} \int_{\mathcal{O}^c} m(z, \hat{z}) \hat{u}(\hat{z}) d\hat{z} \quad (6.16)$$

Człon $n_\Omega(z)$ oznacza wektor normalny do Ω dla $z \in \partial\Omega$.

Dokładne wyprowadzenia zależności można znaleźć w [10].

6.3 Wielostopniowy schemat wypełniania.

W wielu algorytmach wypełniania braków w obrazie stosowany jest algorytm wielostopniowy, przykładem mogą być [29], [30] bądź [31]. Algorytm polega na utworzeniu serii pomniejszych przeskalowanych obrazów, następnie wypełnianiu ich poczynawszy od najmniejszego z warunkami początkowymi uzyskanymi poprzez przeskalowanie do większego obrazu uprzednio wyznaczonych wartości w masce. W przypadku najmniejszego obrazu z serii warunkami początkowymi mogą być:

- losowo wyznaczone wartości obrazu pobierane z obszaru \mathcal{O}^c
- rozwiązanie równania Poisson'a (3.31), dokładnie przedstawione w Rozdział 3, metodą SOR zgodnie z zależnością (3.35).

Przeskalowywanie w obie strony musi odbywać się w sposób nie zniekształcający głównych struktur, obiektów i tekstur obrazu. W tym celu autorzy w [10] proponują model Gauss'owskiej piramidy.

W celach wmalowywania tworzone są dwie oddzielne piramidy wyznaczające maski obrazu i same obrazy. Niech S oznacza liczbę stopni piramidy, a oryginalny wymiar jest oznaczany przez $s = 0$. Niech rozmiar najmniejszego stopnia określa A_{s-1} , natomiast rozmiar oryginalnego obrazu A_0 . Wtedy współczynnik próbkowania do tworzenia coraz mniejszych obrazów wynosi:

$$r := \left(\frac{A_0}{A_{s-1}} \right)^{\frac{1}{s-1}} \quad (6.17)$$

W rezultacie aby uzyskać mniejszy obraz musi zachodzić $r \geq 1$. Parametr σ tworzonego filtru Gauss'a wyznaczany jest z zależności $\sigma(r) = 0.62\sqrt[2]{r^2 - 1}$. Jako rozmiar okna Gauss'a wykorzystywanego do filtrowania można przyjąć jako stały uprzednio definiowany. Niech indeks górny s odnosi się do zmiennych i obszarów w zakresie rozważanego stopnia piramidy. W przypadku tworzenia kolejnych masek wykorzystywane jest próbkowanie w dół z ograniczeniem w postaci zależności między funkcjami charakterystycznymi wmalowywanych obszarów \mathcal{O}^s i \mathcal{O}^{s+1} :

$$\chi_{\mathcal{O}^{s+1}} = (\downarrow_r g_{\sigma(r)} * \chi_{\mathcal{O}^s}) > T_0. \quad (6.18)$$

Autorzy w [10] proponują wartość $T_0 = 0.4$. Niech f stanowi wynik próbkowania w dół z filtrem Gauss'a. Zapis oznacza, iż w przypadku gdy $f(x) > 0.4$ punkt w przekształceniu traktowany jest jako punkt źródłowy obrazu nie wymagający wmalowania. W przypadku obrazu wykorzystywana jest zależność

$$u^{s+1}(x) = \downarrow_r \frac{g * \chi_{(\mathcal{O}^s)^c} u^s(x)}{g * \chi_{(\mathcal{O}^s)^c}(x)} \quad (6.19)$$

Powyższe równanie przedstawia ograniczenie w postaci nieuwzględniania wartości znajdujących się w obszarze \mathcal{O}^c . Obraz w pierwszym kroku zostaje przemnożony przez maskę przedstawiającą źródłową część obrazu, następnie dokonywana jest operacja splotu obrazu z utworzonym filtrem Gauss'a normalizowany członem $g * \chi_{(\mathcal{O}^s)^c}(x)$. W przypadku próbkowania do góry, wyznaczane są dwie piramidy: obraz w ograniczeniu do obszaru wmalowanego w poprzednim stopniu oraz pomocnicza funkcja przedstawiająca mapę podobieństwa. Do tego został wykorzystany algorytm dokładnie przedstawiony w [31] i opisany w [10].

6.4 Podsumowanie algorytmu

Łącząc opisane powyżej operacje dokonywane na obrazie algorytm wypełniania można przedstawić w następujących krokach:

1. Ustawienie argumentów algorytmu: obraz u^0 , maska obrazu χ , liczba poziomów piramidy S , rozmiar najmniejszego obrazu A_{S-1} , ilość iteracji minimalizacji energii I
2. Wyznaczenie piramidy maski stanowiącej zbiór: $\{\chi^s\}_{s=0,\dots,S-1}$
Wyznaczenie piramidy obrazu stanowiącej zbiór: $\{u^s\}_{s=0,\dots,S-1}$
3. Inicjalizacja obrazu u^{S-1} w obszarze \mathcal{O}
Wyznaczenie mapy podobieństwa φ^{S-1} na podstawie u^{S-1}
4. Wykonanie dla każdego z poziomów piramidy s począwszy od $S - 2$ do 0 :
 - a) Przeskalowanie mapy podobieństwa z zastosowaniem algorytmu interpolacji najbliższego sąsiada $\varphi^s = r \cdot (\uparrow_r \varphi^{s+1})$
 - b) Wyznaczenie przeskalowanego obrazu $u_0^s = r \cdot (\uparrow_r u^{s+1})$ i przepisanie wartości $u^s(\mathcal{O}) = u_0^s(\mathcal{O})$ w celu inicjalizacji
 - c) Powtarzanie I -tą ilość razy z wykorzystaniem odpowiedniej normy (6.3)
 - minimalizacja (6.2) w celu wyznaczenia φ^s
 - minimalizacja (6.4) w celu wyznaczenia u^s

Wynikiem powyższej instrukcji jest zbiór obrazów $\{u^s\}$ oraz map podobieństwa $\{\varphi^s\}$, gdzie $s \in 0, \dots, S - 1$

Bibliografia

- [1] Shutao Li and Ming Zhao. Image inpainting with salient structure completion and texture propagation. *Pattern Recognition Letters*, 32(9):1256–1266, 2011.
- [2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/ Addison-Wesley Publishing Co., 2000.
- [3] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.
- [4] Jianhong Shen and Tony F Chan. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002.
- [5] Jinming Duan, Zhenkuan Pan, Wangquan Liu, and Xue-Cheng Tai. Color texture image inpainting using the non local ctv model. *Journal of Signal and Information Processing*, 4(03):43, 2013.
- [6] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *iccv*, page 1033. IEEE, 1999.
- [7] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

- [8] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. In *ACM Transactions on Graphics (ToG)*, volume 24, pages 861–868. ACM, 2005.
- [9] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, 12(8):882–889, 2003.
- [10] Pablo Arias, Gabriele Facciolo, Vicent Caselles, and Guillermo Sapiro. A variational framework for exemplar-based image inpainting. *International journal of computer vision*, 93(3):319–347, 2011.
- [11] A Salih. Streamfunction-vorticity formulation. *Department of Aerospace Engineering Indian Institute of Space Science and Technology*, 2013.
- [12] MA Ebrahimi and Evelyn Lunasin. The navier–stokes–voight model for image inpainting. *The IMA Journal of Applied Mathematics*, 78(5):869–894, 2012.
- [13] Rein van den Boomgaard. Algorithms for non-linear diffusion.
- [14] Dalia Fishelov and Nir Sochen. Image inpainting via fluid equations. In *Information Technology: Research and Education, 2006. ITRE'06. International Conference on*, pages 23–25. IEEE, 2006.
- [15] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.
- [16] Pascal Getreuer. Total variation inpainting using split bregman. *Image Processing On Line*, 2:147–157, 2012.
- [17] Miyoung Jung, Xavier Bresson, Tony F Chan, and Luminita A Vese. Nonlocal mumford-shah regularizers for color image restoration. *IEEE transactions on image processing*, 20(6):1583–1598, 2011.

- [18] Jinming Duan, Zhenkuan Pan, Baochang Zhang, Wanquan Liu, and Xue-Cheng Tai. Fast algorithm for color texture image inpainting using the non-local ctv model. *Journal of Global Optimization*, 62(4):853–876, 2015.
- [19] Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.
- [20] Xue-Cheng Tai, Jooyoung Hahn, and Ginmo Jason Chung. A fast algorithm for euler’s elastica model using augmented lagrangian method. *SIAM Journal on Imaging Sciences*, 4(1):313–344, 2011.
- [21] Junfeng Yang, Wotao Yin, Yin Zhang, and Yilun Wang. A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM Journal on Imaging Sciences*, 2(2):569–592, 2009.
- [22] Peter Blomgren and Tony F Chan. Color tv: total variation methods for restoration of vector-valued images. *IEEE transactions on image processing*, 7(3):304–309, 1998.
- [23] SN LAKSHMI MALLUVALASA. Implementation of patch-based inpainting method.
- [24] Junqing Chen, Thrasyvoulos N Pappas, Aleksandra Mojsilovic, and Bernice E Rogowitz. Adaptive perceptual color-texture image segmentation. *IEEE Transactions on Image Processing*, 14(10):1524–1536, 2005.
- [25] Manik Varma and Andrew Zisserman. A statistical approach to material classification using image patch exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 31(11):2032–2047, 2009.
- [26] F. Masci. Line detection by hough transformation. 2009.
- [27] Luminita A Vese and Stanley J Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of scientific computing*, 19(1-3):553–572, 2003.

- [28] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3):24, 2009.
- [29] Norihiko Kawai, Tomokazu Sato, and Naokazu Yokoya. Image inpainting considering brightness change and spatial locality of textures and its evaluation. In *Pacific-Rim Symposium on Image and Video Technology*, pages 271–282. Springer, 2009.
- [30] Nikos Komodakis and Georgios Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 16(11):2649–2661, 2007.
- [31] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (3):463–476, 2007.

Spis treści

1	Wprowadzenie	2
2	Przegląd istniejących algorytmów.	5
3	Równania różniczkowe wyższego rzędu.	7
3.1	Rozwiązanie równania Naviera-Stokesa	7
3.2	Problem minimalizacji energii	16
3.2.1	Modyfikacje algorytmu	24
4	Uzupełnianie kawałkami obrazu.	27
5	Wmalowywanie struktury i tekstury obrazu.	40
6	Wariacyjny, nielokalny model wmalowywania	46
6.1	Wyznaczenia mapy podobieństwa.	48
6.2	Wyznaczenie wartości obrazu.	49
6.3	Wielostopniowy schemat wypełniania.	52
6.4	Podsumowanie algorytmu	54

Warszawa, dnia

Oświadczenie

Oświadczam, że pracę magisterską pod tytułem: „Poprawa jakości obrazów cyfrowych poprzez metodę wmalowywania”, której promotorem jest dr Sławomir Skonieczny, wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

.....