UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Isabela Góes Rangel

Análise de saliência visual baseada em contraste e em informações de fronteiras

RECIFE
2017

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Isabela Góes Rangel

Análise de saliência visual baseada em contraste e em informações de fronteiras

Monografia apresentada ao Centro de Informática (CIN) da Universidade Federal de Pernambuco (UFPE), como requisito parcial para conclusão do Curso de Engenharia da Computação, orientada pelo professor Carlos Alexandre Barros de Mello.

RECIFE
2017

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Isabela Góes Rangel

Análise de saliência visual utilizando métodos bottom-up

Monografia submetida ao corpo docente da Universidade Federal de Pernambuco, defendida e aprovada em 05 de Dezembro de 2017.

Banca Examinadora:

Orientador

Carlos Alexandre Barros de Mello

Doutor

Examinador

Tsang Ing Ren

Doutor

AGRADECIMENTOS

Agradeço à minha família: pais, avó e marido pela dedicação e presença. Agradeço também aos meus amigos em especial: Bianca Lisle, Maria Luiza, Rodrigo Castiel, Amanda Isabel e Leonardo Santos por sempre estarem perto.

Agradeço ao Professor Carlos Alexandre Barros de Mello pela paciência e excelência na orientação desse trabalho.

RESUMO

A atenção visual é o mecanismo utilizado pelo sistema visual humano para selecionar áreas de interesse em uma cena. Essas áreas podem ser ditas salientes e se destacam, de acordo com alguma classificação, em relação à sua vizinhança. A saliência visual é uma área de grande importância para segmentação de imagens, busca visual e compressão de imagens. Nesse trabalho é realizado um estudo dos principais métodos na área de saliência visual, os quais são comparados com um novo algoritmo baseado em contraste e em informações extraídas das margens da imagem.

Palavras-chave: Atenção visual, Detecção de saliência, Segmentação de Imagens

ABSTRACT

The visual attention is the mechanism used by the Human Visual System to select the areas of most importance in a scene. These areas are called salient and stand out from its neighborhood according to a criterion. Visual saliency is an area of high importance to image segmentation, visual search and image compression. In this work a study of the main methods in the area of visual salience was realized. These methods were compared to a new approach based on contrast and boundary information.

Keywords: Visual attention, Saliency detection, Image segmentation

Sumário

Lista de Figuras

TABELA DE SIGLAS

Sigla	**Significado**
SR	*Saliency Residual*
SIG	*Signature Saliency*
FT	*Frequency Tuned Saliency*
SF	*Saliency Filters*
GBVS	*Graph Based Visual Saliency*
SLIC	*Simple Linear Iterative Clustering*
DCT	*Discrete Cosine Transform*
IDCT	*Inverse Discrete Cosine Transform*

1. INTRODUÇÃO

1.1. MOTIVAÇÃO

Atenção visual é o mecanismo que o sistema visual humano (SVH) utiliza para direcionar o foco a objetos ou regiões de uma cena observada. Essas regiões em destaques são ditas salientes. Saliência intuitivamente caracteriza partes de uma cena, podendo ser objetos ou regiões, que aparentam para um observador se destacar de regiões vizinhas [1]. O sistema visual humano ao observar uma imagem consegue detectar regiões salientes através de combinação de fatores. O primeiro conjunto de fatores contém apenas características da cena observada (como cor, intensidade e orientação). Esse processo (dito *exógeno*) pode ser denominado *bottom-up*. O segundo conjunto de fatores que direciona a atenção visual contém fatores cognitivos (como memória e conhecimento prévio). Esse processo (dito *endógeno*) pode ser denominado *top-down*. Acredita-se que são utilizados tanto os processos *top-down* quanto os *bottom-up* pelo sistema visual humano ao analisar uma cena.

Os modelos computacionais que simulam a atenção visual se baseiam principalmente na abordagem *bottom-up*, pois são mais rápidos que os *top-down*. O primeiro modelo foi proposto por Koch e Ullman. Nele, a partir de características da cena, são construídos mapas de características que convergem para um mapa de saliência final, representando o quanto cada região da cena se destaca [1]. O método que implementa esse modelo foi proposto por Itti, Koch e Niebur [3] e é um dos principais métodos da área (Figura 1).

A área de saliência visual tem aplicações principalmente no pré-processamento de imagens para outras finalidades. Na área de robótica, por exemplo, pode ajudar a selecionar área de interesse. Na segmentação de imagens, ajuda a extrair o *foreground*, parte saliente da imagem. No reconhecimento de objetos, auxilia a extrair informações sobre os objetos que podem servir de entrada para um algoritmo de aprendizagem. Na compressão de imagens, pode indicar a parte mais importante de uma imagem e evitar perdas nessas regiões salientes.

1.2. OBJETIVO

Este trabalho tem como objetivo implementar o método proposto por Xu e Zhang em [4] e comparar os resultados de métodos estado-da-arte: Itti-Koch-Niebur (Itti), *Graph Based*

Visual Saliency (GBVS), *Image Signature* (SIG), *Saliency Filters* (SF) e *Frequency Tuned Saliency* (FT). Todos os métodos serão avaliados na base MSRA disponibilizada por [6].

1.3 ESTRUTURA DO TRABALHO

Para atingir os objetivos, o trabalho está estruturado da seguinte maneira: no Capítulo 2 são apresentados os métodos estado-da-arte; no Capítulo 3 é apresentado o método implementado; no Capítulo 4 são apresentados os resultados obtidos nos experimentos e, no Capítulo 5, a conclusão e proposta de trabalhos futuros.

2. ESTADO DA ARTE

Existem duas categorias principais para os métodos que detectam a saliência visual: *bottom-up* e *top-down*. Na categoria *bottom-up*, a escolha da região saliente é motivada por fatores intrínsecos à imagem como cor, intensidade e orientação. Na categoria *top-down*, fatores cognitivos, como busca por um objeto específico ou finalizar uma tarefa, influenciam na detecção da região saliente. Neste capítulo, são analisados apenas métodos *bottom-up*, uma vez que são mais simples que os *top-down*.

2.1. ITTI- KOCH-NIEBUR

O método proposto por Itti, Koch e Niebur em [3] é um dos trabalhos mais importantes na área de saliência visual. Ele se baseia em um modelo biológico proposto por Koch e Ullman para representar o sistema visual humano. Os passos desse método podem ser visualizados na Figura 1.

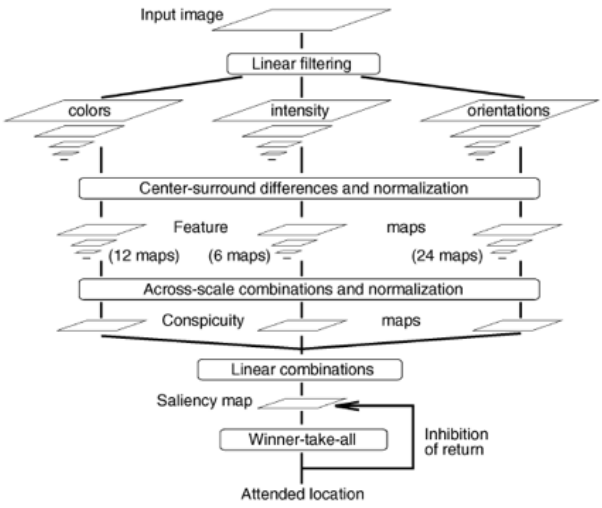

Figura 1: Esquema do Método de Detecção de Mapa de Saliência definido por Itti, Koch e Niebur.

Inicialmente, a imagem é decomposta pelas características de cor, intensidade e orientação. Considerando uma imagem no sistema RGB, com r, g e b sendo os componentes vermelho, verde e azul da imagem original, é calculada uma imagem de intensidade *I*:

$$I = (r + g + b)/3$$

Para representar estímulos de intensidade, I é usada para criar uma pirâmide gaussiana $I(\sigma), \sigma \epsilon [0..8]$ onde σ é a escala. Tal pirâmide é composta por 9 imagens com escalas variando de 1:1 (onde a imagem original é a nível 0) até 1:256 (nível 8).

Para representar estímulos de cor as componentes r, g e b são normalizadas por I para desacoplar a cor da intensidade e são utilizadas para criar 4 canais $R, G, B\ e\ Y$:

$$R = r - \frac{(g + b)}{2}$$

$$G = g - \frac{(r + b)}{2}$$

$$B = b - \frac{r + g}{2}$$

$$Y = r + g - 2(|r - g| + b)$$

Cada um desses canais retorna uma resposta máxima para qual a cor está relacionada e zero para preto ou branco. A partir desses 4 canais, são criadas 4 pirâmides gaussianas R(σ), G(σ), B(σ) e Y(σ), $\sigma \epsilon [0..8]$.

Para representar estímulos de orientação, a partir de I são utilizadas filtros de Gabor para construir as pirâmides $O(\sigma, \theta)$, $\sigma \epsilon [0..8]$ $e\ \theta \epsilon \{0°, 45°, 90°, 135°\}$.

No próximo passo, são gerados 3 conjuntos de mapas de características relacionados a intensidade, cor e orientação que incorporam características do sistema visual humano. O sistema humano visual é mais sensível a estímulos em uma área central ao invés de estímulos em sua vizinhança. Para simular esse mecanismo chamado "*center-surround*", o modelo define o centro como um pixel na escala c = {2,3,4} e sua vizinhança como um pixel na escala $c = s + \delta = \{3,4\}$. Como os mapas obtidos na etapa anterior estão em diferentes escalas é introduzido um operador centro-vizinhança " $\ominus$ " ,onde A $\ominus$ B corresponde a uma interpolação para uma escala mais fina e realizar uma subtração ponto a ponto entre A e B.

O primeiro conjunto desses mapas de características representando os estímulos de intensidade é gerado a partir da pirâmide gaussiana $I(\sigma)$ e é computado de acordo com a equação:

$$I(c, s) = |I(c) \ominus I(s)|$$

Como c ={2,3,4}e $c = s + \delta = \{3,4\}$, as escalas possíveis para $c\ e\ s$ são 2-5, 2-6, 3-6, 3-7, 4-

7, 4-8 , fornecendo 6 mapas de intensidade.

O segundo conjunto de mapas é gerado a partir de R(σ), G(σ), B(σ) e Y(σ), $\sigma \epsilon$ [0..8]. Esses mapas representam a característica do sistema visual humano da oponência cromática entre as cores: neurônios são estimulados no centro por uma cor e inibidos por outra e o contrário acontece nas vizinhanças. Isto acontece para as cores vermelho/verde e verde/vermelho e entre azul/amarelo e amarelo/azul.

$$RG(c,s) = |(R(c) - G(c)) \ominus (G(s) - R(s))|$$

$$BY(c,s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|$$

São gerados 6 mapas RG(c,s) e 6 mapas BY(c,s) totalizando 12 mapas de cor.

O terceiro conjunto de mapas representando a orientação é gerado a partir de $O(\sigma, \theta)$ totalizando 24 mapas de orientação:

$$O(c,s,\theta) = |O(c,\theta) \ominus O(s,\theta)|$$

A fim de combinar mapas que representam grandezas diferentes, um operador de normalização $\mathcal{N}(\cdot)$ é introduzido, onde mapas que possuem pequena quantidade de picos(locais salientes) são promovidos e os que tem resposta de intensidade similar são suprimidos. $\mathcal{N}(\cdot)$ é definido como:

(i) Normalizar todos valores do mapa para [0..M]

(ii) Achar localização do máximo global M e computar média $\overline{m}$ entre regiões de máximo local

(iii) Multiplicar o mapa por $(M - \overline{m})^2$

Para cada conjunto de mapas de características relacionados à intensidade, cor e orientação gerados anteriormente é gerado um único mapa $\bar{I}, \bar{C}$ e $\bar{O}$:

$$\bar{I} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{c+4} \mathcal{N}(I(c,s))$$

$$\bar{C} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{c+4} \left[\mathcal{N}\left(RG(c,s)\right) + \mathcal{N}\left(BY(c,s)\right)\right]$$

$$\bar{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} \mathcal{N}\left(\oplus_{c=2}^{4} \oplus_{s=c+3}^{c+4} \mathcal{N}(O(c,s,\theta))\right)$$

Sendo $\oplus$ uma operação que reduz as imagens para escala $\sigma = 4$ e realiza uma soma ponto a ponto. O mapa de saliência final pode ser calculado usando:

$$\mathcal{S} = \frac{1}{3}(\mathcal{N}(\bar{I}) + \mathcal{N}(\bar{C}) + \mathcal{N}(\bar{O}))$$

Devido à criação das pirâmides gaussianas o método se torna computacionalmente custoso.

A partir de experimentos realizados é possível verificar que as bordas das figuras salientes não estão definidas, como mostrado na Figura 2.

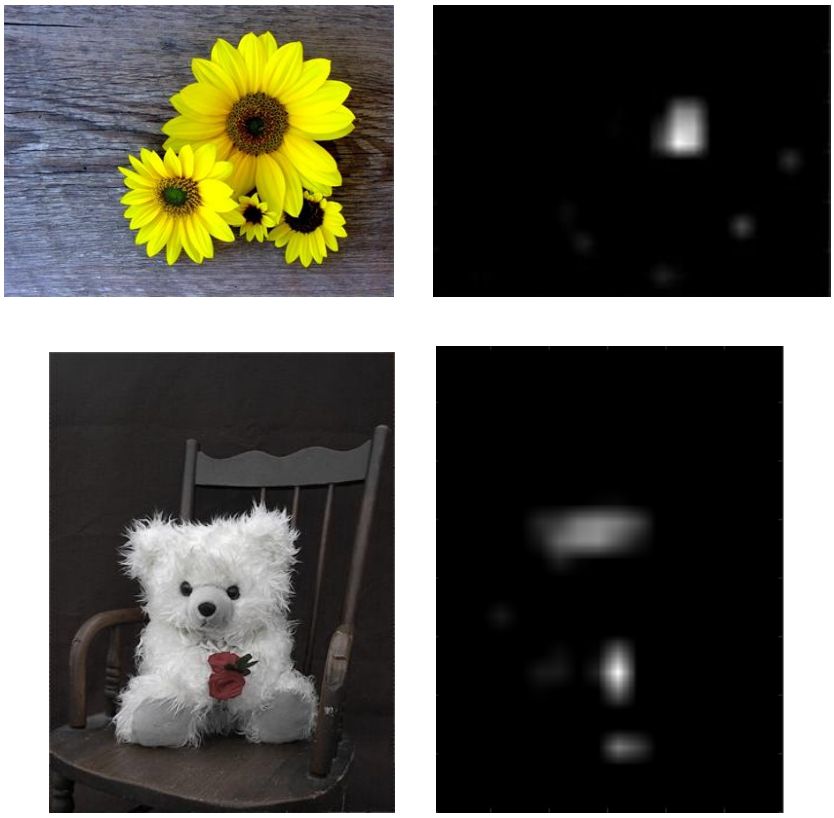

Figura 2: Imagem original e mapa de saliência obtido utilizando método proposto por Itti

2.2. GRAPH BASED VISUAL SEARCH (GBVS)

De acordo com o autor do método *Graph Based Visual Search* proposto em [5], os principais modelos de saliência visual podem ser descritos em 3 etapas:

1) Extração: obtenção de mapas de características no plano da imagem;

2) Ativação: formação de um "mapa de ativação" (ou mais de um) utilizando a etapa anterior;

3) Normalização: normalização dos mapas de ativação seguido da unificação em um mapa de saliência.

O algoritmo usado por Itti-Koch-Niebur [2] utiliza todas as etapas descritas acima. Primeiro ocorre a extração de características usando filtragem (cor, intensidade e orientação). Depois são construídos conjuntos de mapas de características utilizando operações de center-surround e normalização. No final os mapas da etapa anterior são unificados para gerar o mapa de saliência.

O GBVS é um modelo bottom-up para as etapas de Ativação e Normalização e calcula a parte da etapa de Extração como no método de Itti-Koch-Niebur.

2.3.1 Etapa de Ativação:

Primeiramente é assumido como entrada os mapas de características *M* da etapa *s1*, que não é calculado pelo GBVS, onde $M: [n]^2 \rightarrow \mathbb{R}$, onde n = [1..n] (*M* é uma matriz *n*x*n* para simplificação, podendo ser utilizado um mapa retangular).

Dada uma imagem *I*, o objetivo é destacar as regiões de maior importância de acordo com algum critério, *e.g.*, fixação do olho humano. Na etapa de ativação *s2* são utilizados os mapas de características M dados como entrada. O objetivo da etapa *s2* é calcular um mapa de ativação *A*, com $A: [n]^2 \rightarrow \mathbb{R}$, onde uma posição (i,j) $\in [n]^2$ na matriz *A* tenha um valor alto se *M(i,j)* é de alguma maneira distinto em sua vizinhança. Essa distinção é feita utilizando um conceito de dissimilaridade. A dissimilaridade de pixels *(i,j)* e *(p,q)* é dada por:

$$d((i,j) \parallel (p,q)) \triangleq \left|\log \frac{M(i,j)}{M(p,q)}\right|$$

É construído um grafo completo e direcionado G_A utilizando *M*. Cada nó G_A é um pixel de M e está ligado a todos os *n-1* nós. O peso de um nó (i,j) ligado a (p,q) tem peso w_1 proporcional a dissimilaridade entre eles:

$$w_1\big((i,j),(p,q)\big) \triangleq d((i,j) \parallel (p,q)) \,.\, F(i-p, j-q)$$

onde:

$$F(a,b) \triangleq e^{\left(-\frac{a^2+b^2}{2\sigma^2}\right)}$$

Com σ sendo um parâmetro livre e variando entre $\frac{1}{10}$ e $\frac{1}{5}$ da largura do mapa. *F* descreve a proximidade entre *(i,j)* e *(p,q)*, sendo maior quando a distância entre os nós é menor. Normalizando os pesos dos vértices de saída de cada nó para 1 e relacionando nós a estados e pesos a probabilidades, podemos fazer uma associação de G_A para uma cadeia de Markov. A distribuição de equilíbrio dessa cadeia implica em acúmulo de massa nos nós que tem maior dissimilaridade (mais distintos) que sua vizinhança. Esse resultado é uma medida de ativação perimitindo detectar as áreas mais salientes. Na Figura 3 percebe-se que o GBVS detecta as áreas mais precisamente que o método de Itti porém não salienta os objetos uniformemente.

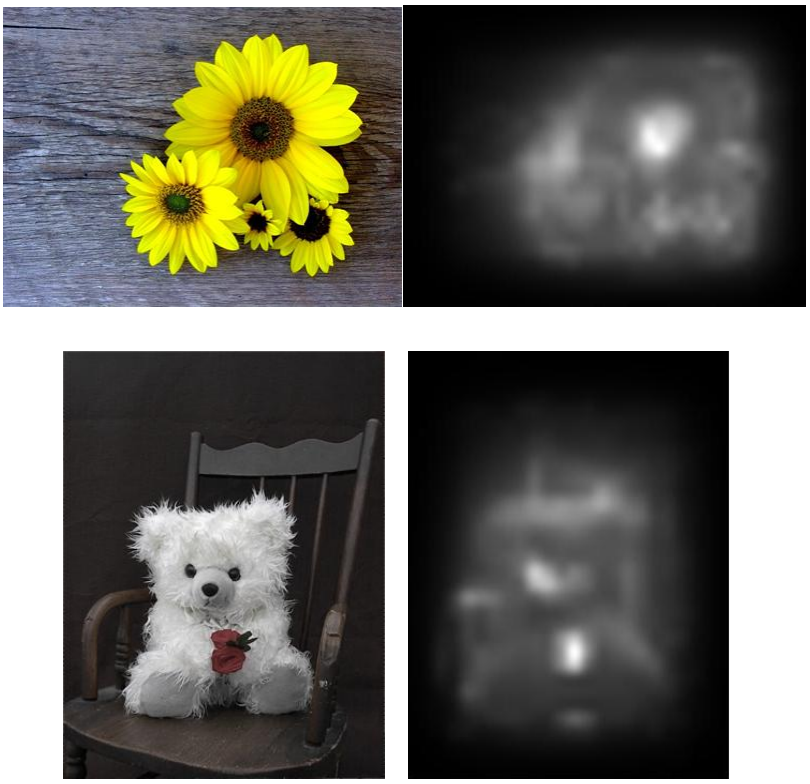

Figura 3: Imagem original e mapa de saliência obtido utilizando método GBVS

2.3. *FREQUENCY TUNED (FT)*

O método descrito por Achanta [6] propõe cobrir falhas de outros métodos da área: regiões com baixa resolução, bordas mal-definidas, alto custo computacional ou não salientar todo objeto de maneira uniforme.

Na Figura 4 é possível observar que o método GBVS [5] teve as bordas mal-definidas, o método proposto por Itti-Koch-Niebur [3] gera mapas com baixa resolução ou não salientea o objeto de maneira uniforme, similar ao método SR (*Spectral Residual*) [11].

O modelo proposto fornece um mapa com alta resolução, com limites bem definidos, destaca uniformemente a região saliente e é computacionalmente eficiente. Os autores propõem cinco requisitos que mapas de saliência devem cumprir:

- Destacar os maiores objetos
- Destacar uniformemente as regiões salientes
- Estabelecer bordas bem-definidas dos objetos salientes
- Desprezar altas frequências relacionadas à textura e ruído
- Produzir mapas de alta resolução

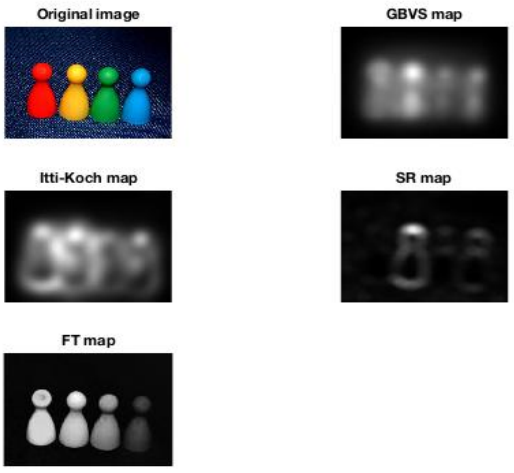

Figura 4: Imagem original e mapa de saliência obtido utilizando os métodos: GBVS[5], Itti[3], SR[11], FT[6].

O mapa de saliência *S* para uma imagem *I* é calculado como:

$$S(x,y) = ||\boldsymbol{I}_{\mu} - \boldsymbol{I}_{\omega_{hc}}(x,y)||$$

onde $\boldsymbol{I}_{\mu}$ é o vetor médio de características no sistema de cores Lab. $\boldsymbol{I}_{\omega_{hc}}$ é o resultado da filtragem de I por um filtro gaussiano 5x5 e ||.|| é a distância Euclidiana. Na Figura 2 é

possível observar que resultados semelhantes são obtidos com $\boldsymbol{I}_{\mu}$ calculado sendo a média de $\boldsymbol{I}_{\omega_{hc}}$ (letra c) em vez de a média de *I* (letra b).

O método proposto cumpre os cinco requisitos para um mapa de saliência e é mais simples para implementar e computacionamente eficiente. Na Figura 6, é possível ver que o objeto saliente foi destacado em relação ao *background*. Em experimentos realizados nesta monografia. foi possível perceber que para imagens em que os objetos salientes são maiores que o *background*, o *background* (ou parte dele) pode ser detectado como saliente como visto na Figura 7.

Figura 5: Imagem original e Mapa de saliência obtido utilizando a média de $\boldsymbol{I}_{\omega_{hc}}$ (b) e imagem original(c) para calcular $\boldsymbol{I}_{\mu}$ (vetor médio de características)

Figura 6: Imagem original e Mapa de saliência obtido utilizando o método FT.

Figura 7: Imagem original e Mapa de saliência obtido utilizando o método FT

2.4. *SALIENCY FILTERS (SF)*

O método proposto em [8] é baseado em duas medidas de contraste: Unicidade e Distribuição Espacial. Primeiro, a imagem é decomposta utilizando o método SLIC (*Simple Linear Iterative Clustering*) [10]. Essa decomposição da imagem em superpixels resulta em uma representação na qual elementos homogêneos que detém características semelhantes (como cor) agrupados juntos preservando estruturalmente a imagem e abstrai detalhes desnecessários.

A Unicidade quantifica o quanto um elemento (superpixel) é diferente de todos os outros, ou seja, uma medida de raridade do elemento. A Unicidade de um superpixel i é definida como:

$$U_i = \sum_{j=1}^{N} \left\| c_i - c_j \right\|^2 \times w(p_i, p_j)$$

onde *N* é a quantidade de superpixels da imagem, c_i é a cor do superpixel *i* no sistema de cores LAB, $\| \, . \, \|$ é a distância Euclidiana, p_i é a posição do superpixel *i* e *w* é uma função que representa uma medida local e controla a influência espacial e é definida como:

$$w(p_i, p_j) = \frac{1}{Z_i} \times e^{-\frac{1}{2\sigma_p{}^2}\left\| p_i - p_j \right\|^2}$$

onde σ controla o alcance da unicidade e Z_i garante que $\sum_{j=1}^{N} w\,(p_i, p_j) = 1$. A distribuição espacial representa o quanto elementos estão compactos, ou seja, agrupados em uma regiões próximas da imagem. A distribuição é definida como:

$$D_i = \sum_{j=1}^{N} \left\| p_j - \mu_j \right\|^2 \times w(c_i, c_j)$$

e $w(c, c_j) = \frac{1}{Z_i} \times e^{-\frac{1}{2\sigma_c{}^2}\left\| c_i - c_j \right\|^2}$ representa a similaridade entre as cores dos superpixels c_i e c_j, e σ_c controla a sensibilidade de cores da distribuição. μ_i é a média ponderada da posição da cor c_i : $\mu_i = \sum_{j=1}^{N} w\,(p_i, p_j) \times p_j = 1$.

A saliência final para cada superpixel é calculada a partir de U_i e D_i com valores normalizados entre [0..1]. Seja k um fator de escala da exponencial com valor 6, de acordo com [8]:

$$S_i = U_i \times e^{(-k \times D_i)}$$

Na Figura 8, pode ser observada uma aplicação desse método para uma imagem com *foreground* bem distinto do *background*.


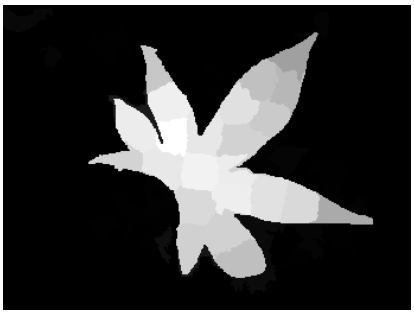

Figura 8: Imagem original e Mapa de saliência obtido utilizando o método SF

2.5. IMAGE SIGNATURE (SIG)

No trabalho descrito em [7], os autores propõem um método para gerar mapas de saliência usando a Transformada Discreta do Cosseno (DCT – *Discrete Cosine Transform*). Assumindo que a imagem de entrada I em tons de cinza seja da forma:

$$I = f + b\,, \qquad I, f, b \in \mathbb{R}^N$$

onde *f* é o *foreground* da imagem e *b* é o *background* da imagem, sendo *f* considerado espacialmente esparso e *b* considerado esparso no domínio da DCT. Esparso significa ter uma peq[1]uena quantidade de elementos não-nulos.

Para detectar o objeto saliente é preciso determinar *f*. Dado I, separar *f* e *b* é um problema difícil, mas para detectar o *foreground* é suficiente detectar os elementos não-nulos de *f*.

Então, dada uma imagem de entrada, ela é levada para o domínio da frequência pela DCT, é aplicada a função *sign(.)*[1] e retorna ao domínio do espaço pela IDCT (Inverso da DCT). Esses passos detectam elementos de interesse (elementos não-nulos) de *f*, calculando a imagem reconstruída $\bar{I}$ e são representados de acordo com a equação:

$$\bar{I} = IDCT[sign(\hat{I})]$$

[1] $\text{sign(x)} = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$

onde $\widehat{I} = DCT(x)$ e a assinatura de uma imagem é definida como:

$$ImageSignature(I) = sign(DCT(I))$$

Para gerar os mapas de saliência m, é utilizado o seguinte cálculo:

$$\boldsymbol{m} = g * (\bar{\boldsymbol{I}} \circ \bar{\boldsymbol{I}})$$

onde g é um filtro gaussiano utilizado para suprimir a quantização causada pela função *sign(.)*, * é o operador de convolução e ∘ é o produto elemento-a-elemento entre matrizes. O desvio padrão do filtro deve ser escolhido proporcionalmente ao tamanho do objeto de interesse, no paper [7] é escolhido como 10% da largura da imagem.

São usados os sistemas de cores Lab e RGB, de forma que dois mapas de saliência são gerados. No sistema RGB, para cada canal de cores é gerado um mapa de saliência e ao final os três são somados como visto na Figura 9. A partir de experimentos realizados, percebe-se que se o *background* é homogêneo, ou tem tonalidades próximas, o algoritmo consegue distinguir bem o *foreground* (Figura 10).

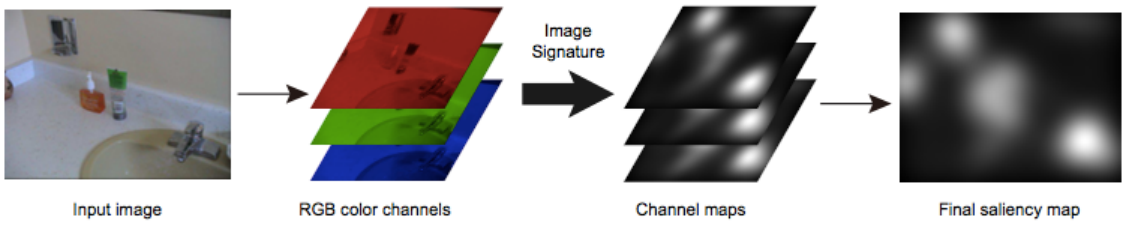

Figura 9: Etapas para gerar mapa de saliência utilizando sistema de cores RGB. (Imagem retirada de [7])

Figura 10– Mapas de saliência gerados utilizando o método SIG para imagem com background homogêneo. (a) Imagem original, (b)Mapa de saliência obtido utilizando LAB (c) Mapa de saliência obtido utilizando RGB.

Para a Figura 11, no sistema de cores RGB a flor não ficou tão saliente quanto no sistema de cores LAB, uma vez que são gerados três mapas e o peso do vermelho na soma final é menor. Também na Figura 10 (b), no SIG utilizando LAB, e na Figura 9 (b) percebe-se que o objeto saliente não é destacado de forma homogênea, uma vez que no interior deles tem tons de preto.

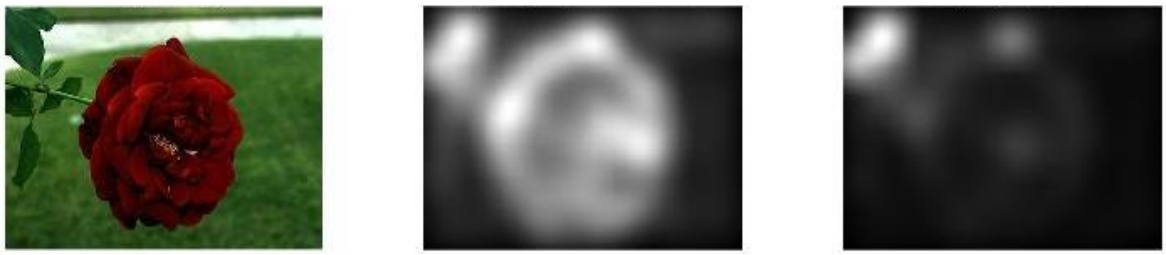

Figura 11 – Mapas de saliência gerados utilizando o método SIG para imagem com objeto saliente vermelho. (a) Imagem original (b) Mapa de saliência obtido utilizando LAB (c) Mapa de saliência obtido utilizando RGB

Em cenários complexos, onde o *background* não é homogêneo, como na Figura 12, o SIG consegue não distinguir o formato do *foreground*.

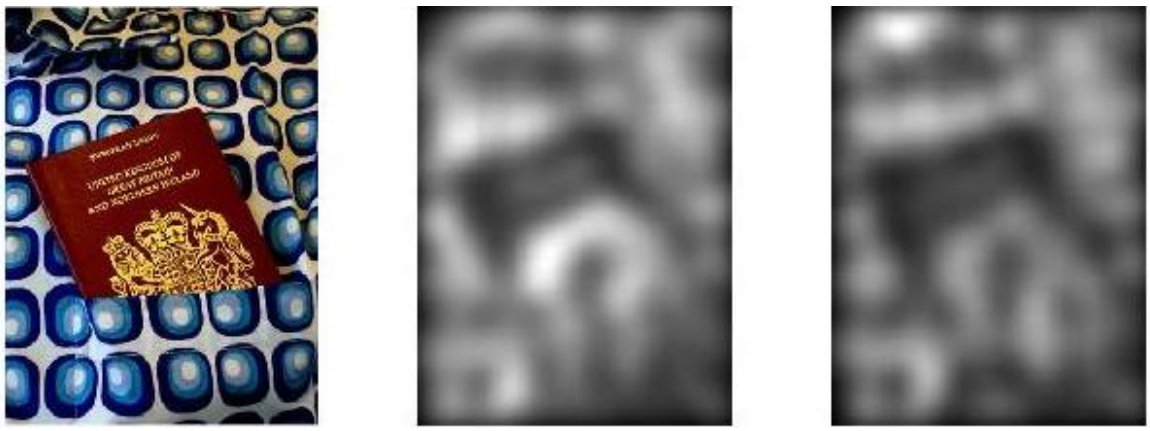

Figura 12 – Mapas de saliência gerados utilizando o método SIG para imagem com background complexo. (a) Imagem original (b) Mapa de saliência obtido utilizando LAB (c) Mapa de saliência obtido utilizando RGB.

3. MÉTODO EM ESTUDO

Neste capítulo, é explicado o método de Xu e Zhang descrito em [4] e implementado neste trabalho. O método consiste em uma abordagem baseada no contraste de cores e nas bordas das imagens.

3.1. VISÃO GERAL

Xu e Zhang propuseram um método *bottom-up* de duas etapas onde a saliência é definida como contraste de cores entre elementos da imagem e elementos das margens da imagem. Na primeira etapa, é construído um mapa de saliência a partir das quatro margens da imagem (superior, inferior, esquerda e direita) e, na segunda etapa, esse mapa é refinado, utilizando uma função de energia. A partir da primeira etapa, já é possível visualizar o objeto saliente e pode ser usado para segmentação de imagens.

3.2. PRIMEIRA ETAPA

Inicialmente, a imagem é divida em superpixels, utilizando o algoritmo SLIC (*Simple Linear Iterative Clustering*) proposto em [10]. A saliência de um superpixel é definida como seu contraste de cor com os superpixels pertencentes às margens da imagem. São utilizados superpixels nas margens ao invés das vizinhanças do superpixel para calcular o contraste de cor, pois utilizar a vizinhança causaria um destacamento das bordas em vez de destacar uniformemente o objeto saliente como visto na Figura 13.

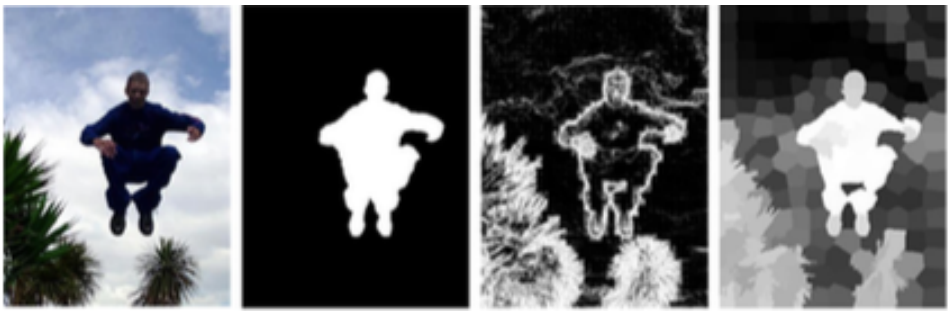

Figura 13 – Imagem original, *Ground truth*, mapa de saliência obtido utilizando superpixels da vizinhança, mapa de saliência obtido utilizando os superpixels nas bordas da imagem.

Com essa definição de saliência, é possível calcular a saliência considerando todas as bordas da imagem integralmente. Para um superpixel *i*, sua saliência *s* é:

$$s(i) = 1 - \frac{1}{k}\sum_{j=1}^{N} w_{ij} \qquad (3.1)$$

onde *N* é a quantidade de superpixels da imagem, *k* é a quantidade de superpixels nas quatro margens da imagem e w_{ij} é o contraste de cores entre os superpixels *i* e *j* e é definido como:

$$w_{ij} = e^{-\frac{\|c_i - c_j\|}{\sigma^2}} \qquad (3.2)$$

onde c_i e c_j representam o valor médio das cores de todos os pixels dentro de um superpixel no sistema de cores LAB e σ é uma constante que controla a importância da distância. O resultado dessa etapa pode ser visto na Figura 12b. Para suprimir o *background* da imagem, a saliência pode ser calculada, considerando cada margem da imagem separadamente. A saliência para um superpixel *i* em relação aos superpixels presentes na margem superior é:

$$s_t = 1 - \frac{1}{k}\sum_{j=1}^{N} w_{ij} \quad i \in 1,2 \ldots N \quad (3.3)$$

onde *k* é a quantidade de superpixels presentes na margem superior. O cálculo da saliência para as margens inferior, direita e esquerda por ser feita de maneira similar. Após essa etapa, os quatro mapas são agrupados em um único chamado *coarse saliency map* de acordo com a equação:

$$s = s_t \times s_b \times s_l \times s_r \quad (3.4)$$

Essa abordagem reduz a imprecisão em casos particulares onde o objeto está em uma das margens da imagem. Na Figura 14c, percebe-se que o *background* foi suprimido com mais intensidade que da 14b, porém ainda não desapareceu completamente.

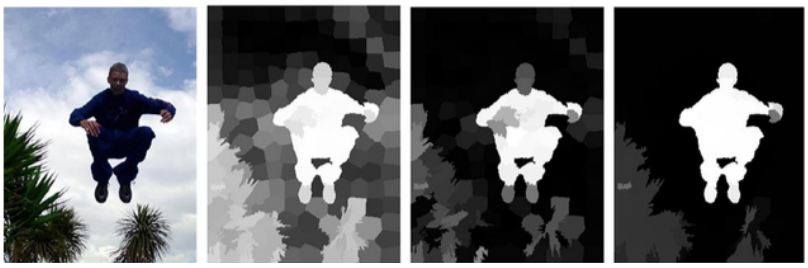

Figura 14 – (a) Imagem original, (b) mapa de saliência obtido utilizando quatro margens integralmente, (c) mapa de saliência obtido utilizando cada margem da imagem separadamente (*coarse saliency map*) e (d) mapa de saliência final. (Imagem retirada de [4])

3.3. SEGUNDA ETAPA

Nessa etapa, o *coarse Saliency map* obtido na etapa anterior é refinado. É observado que um superpixel é relevante para superpixels vizinhos e também para vizinhos dos vizinhos, levando regiões vizinhas a terem valores de saliência similares. Baseado no *coarse Saliency map* e em propriedades de suavização local (*local smoothness*) uma função de energia é proposta:

$$v^* = argmin_s \frac{1}{2}\left(\sum_{i,j=1}^{N} w_{ij}(v_i - v_j)^2 + \mu \sum_{i=1}^{N} \phi_i(v_i - s_i)^2 + \lambda \sum_{i=1}^{N} T_i(v_i - z_i)^2\right) \quad (3.5)$$

No primeiro somatório, $v = [v_1, \dots, v_N]^T$ representa a saliência de cada superpixel, w_{ij} é o peso de dois superpixels interligados o que é o mesmo que a distância de cores entre eles. Essa parte da equação representa que a saliência não deve variar muito entre superpixels vizinhos.

No segundo somatório, s_i é o valor da saliência de um superpixel *i* no *coarse Saliency map*. ϕ_i é a função:

$$\phi_i = e^{\frac{-s_i(1-s_i)}{{\sigma_1}^2}} \quad (3.6)$$

Com essa função, se um superpixel tem valor mais próximo de 1 ou 0 no *coarse Saliency map*, ele tem mais chance de pertencer ao *foreground* ou *background* e um maior impacto no resultado final. A constante μ controla a importância do *coarse Saliency map*; quanto maior for μ, maior o peso do segundo somatório na função de energia.

No terceiro somatório, λ é uma constante que controla a importância desse terceiro termo. Inspirado pelo método descrito em [12], um *thresholding* é realizado a fim de obter os superpixels pertencentes ao *foreground* ou *background* chamados de T_i. Sejam α e β constantes e *M* a média do *coarse Saliency map:*

$$T_i = \begin{cases} 1, & se\ s_i \geq \alpha M \\ 1, & se\ s_i < \beta M \\ 0, & caso\ contrário \end{cases} \quad (3.7)$$

Se para um superpixel *i* sua saliência $s_i \geq \alpha M$, ele pertence ao *foreground*. Se para um superpixel *i* sua saliência $s_i < \beta M$, ele pertence ao *background*. Este somatório e o segundo somatório representam o fato que o mapa de saliência final deve ser semelhante ao *coarse Saliency map.*

A solução para a função de energia é:

$$v^* = (D - W + \mu\psi + \lambda T)^{-1}(\mu\psi S + \lambda T z) \quad (3.8)$$

onde W = $[w_{ij}]_{N\times N}$ é a matriz de relevância de cores, D = $diag\ \{d_{11}, \ldots, d_{NN}\}$ é uma matriz diagonal e d_{ii} é a soma do vetor coluna *i* da matriz de cores W, $d_{ii} = \sum_j w_{ij}$. $\psi = diag\{\phi_i, \ldots, \phi_N\}$ é uma matriz diagonal, T = $diag\{T_1, \ldots, T_N\}$ representa os pixels do *foreground* e *background*. S = $[s_1, \ldots, s_N]^T$ é o valor da saliência do *coarse Saliency map* obtido na seção anterior e z = $[z_1, \ldots, z_N]^T$ são os superpixels do *foreground*.

4. EXPERIMENTOS E ANÁLISE

Este capítulo apresenta os experimentos realizados para comparar o método de Xu e Zhang implementado em Matlab e apresentado no Capítulo 3 com os métodos apresentados no Capítulo 2. Para a comparação, os mapas de saliência são segmentados utilizando um *threshold* que varia de 0 a 255 e são calculados valores de *precison* P e *recall* R:

$$P = \frac{T_p}{T_p + F_p} \qquad R = \frac{T_p}{T_p + F_n}$$

onde T_p são os verdadeiros positivos, F_p são os falsos positivos e F_n são os falsos negativos. P permite mensurar a quantidade de informação detectada que é relevante e R permite mensurar a quantidade de informação relevante detectada. Com esses valores é possível gerar curvas de *precision-recall* que representam o quão bem varios mapas de saliência detectam regiões salientes nas imagens [6]. Para todos os experimentos, foi utilizada a base MSRA-1000 com *ground-truth* obtidos por [6]. As imagens da base foram obtidas em [14] e os *ground-truth* em [15]. A base contém 1000 imagens JPG, no sistema de cores RGB com dimensões 400x300 ou 300x400. A Figura 15 mostra algumas imagens dessa base.

Figura 15 – Imagens da base MSRA-1000

4.1 EXPERIMENTO 1

Para comparar todos os métodos Itti, GBVS, FT, SIG, SF e o método de Xu e Zhang, foram obtidos os mapas de saliência dos métodos Itti, GBVS, FT, SIG, SF utilizando códigos disponibilizados pelos autores.

Para o método de Xu e Zhang foi feita uma implementação em Matlab. Com os parâmetros especificados em [4] não foi possível obter mapas de saliência. Os parâmetros utilizados na implementação foram obtidos empiricamente. Para a primeira etapa foi utilizado σ=30 (Equação 3.1). Para a segunda etapa: λ=1(Equação 3.5), μ=1.1(Equação 3.5), α=2.65 (Equação 3.7) e β=0.3(Equação 3.7). O número de superpixels utilizado foi N=200.

A curva de *precision-recall* obtida está na Figura 16 contém a curva de *precision-recall* comparando os métodos Itti, GBVS, FT, SIG, SF e o método de Xu e Zhang. A Figura 17 contém a curva de *precision-recall* comparando os métodos Itti, GBVS, FT, SIG, SF e o método de Xu e Zhang com resultados obtidos após a primeira etapa (utilizando a Equação 3.4).

Para ambas figuras, é possível perceber que o método SF e o método de Xu e Zhang (na Figura 16 representado pela legenda 'CC') tiveram resultados superiores aos outros. O pior desempenho foi do método SIG.

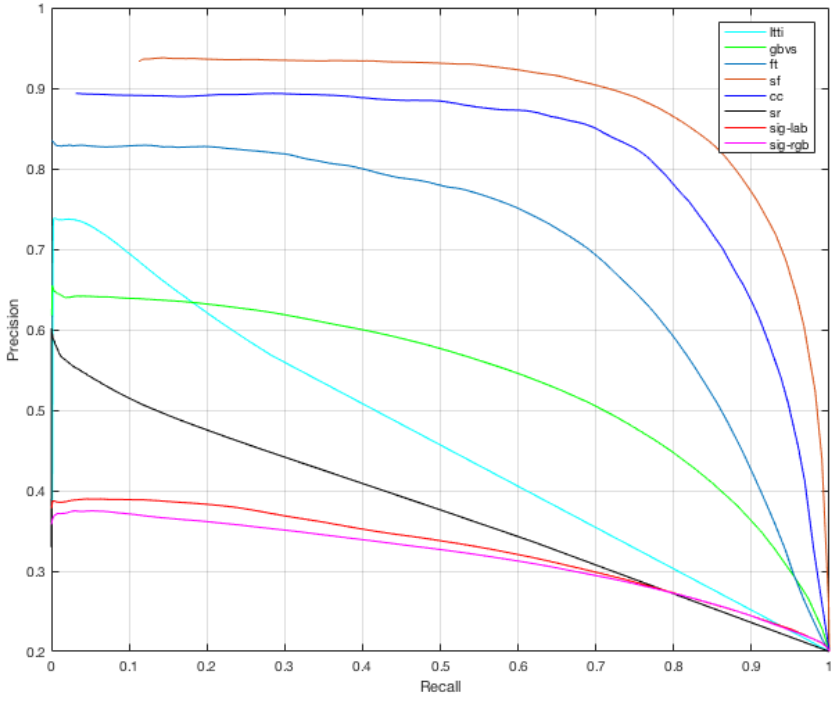

Figura 16 – Curva de *precision-recall* utilizando a base MSRA-1000. CC representa o método implementado no capítulo 3.

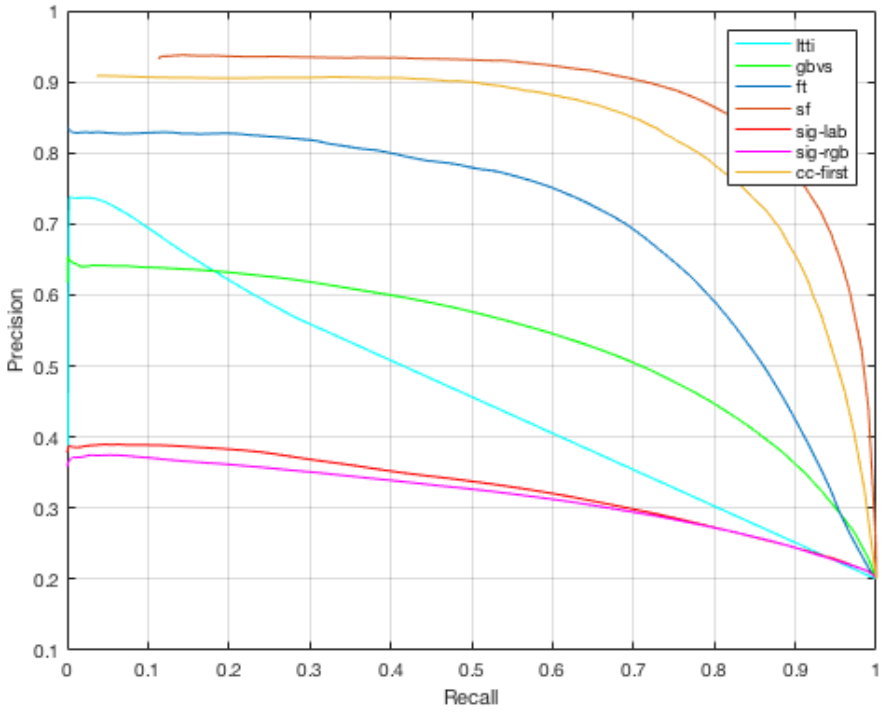

Figura 17 – Curva de *precision-recall* utilizando a base MSRA-1000. CC representa o método implementado no capítulo 3 utilizando resultados do primeiro estágio.

4.3 EXPERIMENTO 2

Foram feitas curvas de *precision-recall* para avaliar o método implementado no capítulo 3 com os mesmos parâmetros utilizados no Experimento 1.

Na Figura 18, foi variado o número de superpixels N (N=50, N=100, N=200, N=400) em que a imagem de entrada era segmentada utilizando o algoritmo SLIC[10]. Quanto maior o número de superpixels há uma queda no desempenho do método.

A Figura 19 apresenta duas curvas. A curva denominada 'CC First Stage' foi obtida utilizando os mapas de saliência obtidos a partir da primeira etapa (utilizando a Equação 3.4), sem utilizar a função de energia descrita na segunda etapa (Seção 3.3). A outra curva denominada 'CC' representa o resultado do método completo (utilizando a primeira e segunda etapa). Há uma queda de desempenho no método quando se utiliza a segunda etapa.

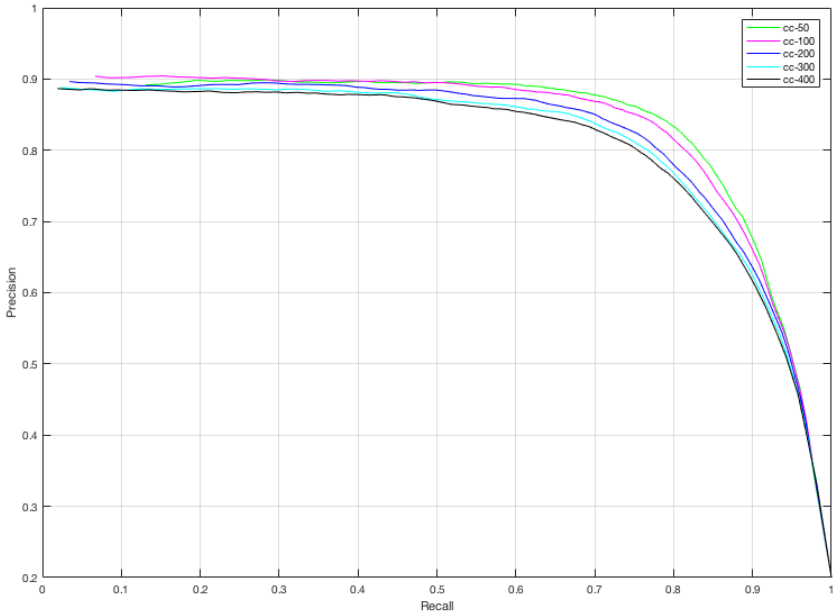

Figura 18 – Curva de *precision-recall* utilizando a base MSRA-1000 variando o número de superpixels N.

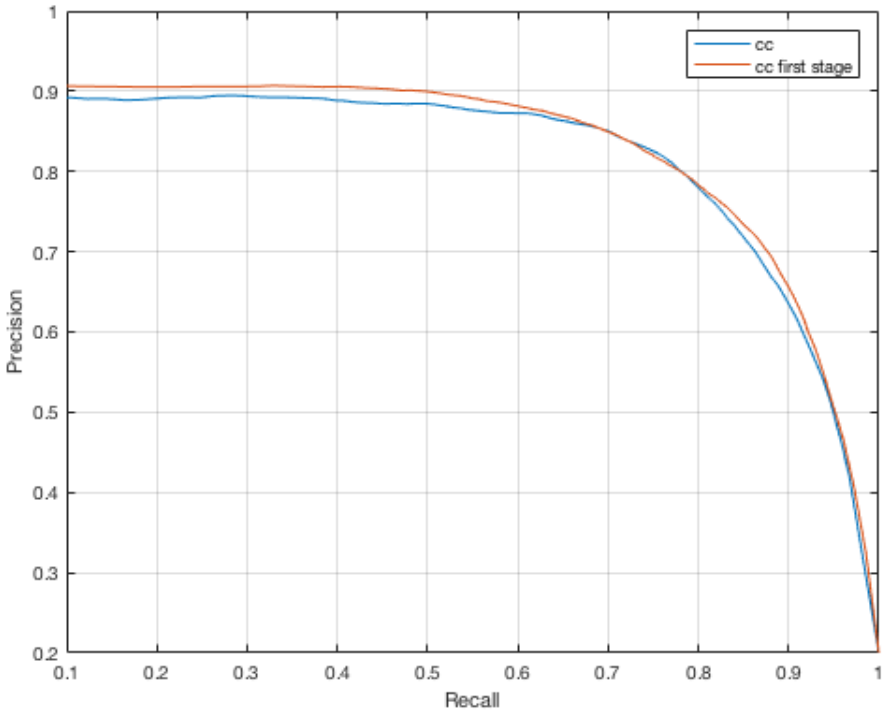

Figura 19 – Curva de *precision-recall* utilizando a base MSRA-1000 comparando os resultados do método a partir do primeiro estágio com os resultados finais.

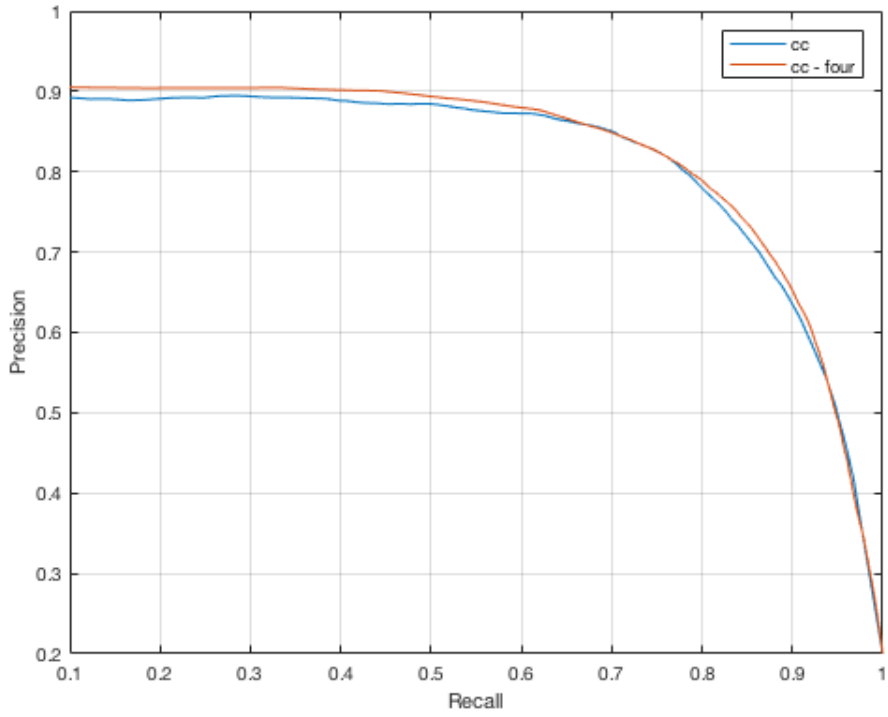

Figura 20 – Curva de *precision-recall* utilizando a base MSRA-1000 comparando os resultados do método com a saliência calculada considerando cada margem individualmente ('cc') e considerando as margens integralmente ('cc-four').

Na Figura 20, a curva "cc – four " representa o método de Xu e Zhang utilizando como definição de saliência a Equação 3.1, que considera as quatro margens da imagem integralmente. A outra curva, "cc" representa o método tradicional, que considera cada margem da imagem individualmente. Também percebe-se que o desempenho do método completo, o qual utiliza a segunda etapa, é ligeiramente menor.

Na Figura 19 e Figura 20 e também na Figura 16 e Figura 17, é possível perceber que há uma queda de desempenho do método após a segunda etapa, o que está relacionado aos parâmetros terem sido obtidos empiricamente, uma vez que os parâmetros fornecidos em [4] não produziram mapas de saliência.

5. CONCLUSÕES E TRABALHOS FUTUROS

5.1 CONCLUSÃO

O objetivo desse trabalho foi comparar os métodos Itti-Koch-Niebur (Itti), *Graph Based Visual Saliency* (GBVS), *Image Signature* (SIG), *Saliency Filters* (SF) e *Frequency Tuned Saliency* (FT) e o método de Xu e Zhang. Para tal, o algoritmo de Xu e Zhang foi implementado em Matlab, enquanto, para os outros métodos, foram usadas implementações disponibilizadas pelos autores. Todos os métodos foram avaliados sobre a base MSRA. O melhor desempenho foi do método SF, seguido do método implementado. O pior desempenho foi do método SIG, tanto utilizando sistema de cores RGB quanto utilizando sistema de cores LAB.

5.2 TRABALHOS FUTUROS

É possível apontar como trabalho futuro a otimização no código do método implementado a fim de obter resultados mais próximos com os descritos em [4]. Este trabalho abordou apenas métodos de detecção de saliência com únicos objetos salientes, a detecção de regiões contendo múltiplos objetos salientes, como descrito em [13], seria uma futura abordagem a ser estudada.

6. BIBLIOGRAFIA

[1] Borji, A.; Itti L.; "State-of-the-Art in Visual Attention Modeling", IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 35, 2013.

[2] Frintrop S.; Rome E.; Christensen H.; "Computational Visual Attention Systems and their Cognitive Foundations: A Survey", ACM Transactions on Applied Perception (TAP), v. 7, 2010.

[3] Itti, L.; Koch, C.; Niebur, E.; "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 20, n.11, p.1254–1259, 1998.

[4] Xu M.; Zhang H.; "Saliency detection with color contrast based on boundary information and neighbors", The Visual Computer (2015), v. 31, 355–364.

[5] Harel, J.; Koch, C.; Perona, P.; "Graph-Based Visual Saliency", Advances in Neural Information Processing Systems 19 (NIPS), p. 545-552, 2006.

[6] R. Achanta; S. Hemami; F. Estrada and S. Susstrunk; "Frequency tuned Salient Region Detection," IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009), p. 1597–1604, 2009.

[7] Hou, X.; Harel, J.; Koch, C. "Image Signature: highlighting sparse salient regions", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 34, n.1, p.194–201, 2012.

[8] Federico Perazzi; Philipp Krähenbül; Yael Pritch; Alexander Hornung; Saliency Filters: Contrast Based Filtering for Salient Region Detection. IEEE CVPR, Providence, Rhode Island, USA, June 16-21, 2012

[9] Cheng, MSRA10K Salient Object Database, 28/07/2014, 22/08/2017 http://mmcheng.net/msra10k/

[10] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk, SLIC Superpixels, EPFL Technical Report 149300, June 2010.

[11] Hou, Xiaodi & Zhang, Liqing. (2007). Saliency Detection: A Spectral Residual Approach. IEEE Conference in Computer Vision and Pattern Recognition, CVPR 2007.

[12] Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 228–242 (2008)

[13] Oh, KangHan & Lee, Myungeun & Kim, Gwangbok & Kim, SooHyung. (2016). Detection of Multiple Salient Objects through the Integration of Estimated Foreground Clues:. Image and Vision Computing. 54, 2016.

[14] https://github.com/bschauerte/SalientObjectsAchanta. Acesso em 22 de Novembro de 2017.

[15] http://ivrl.epfl.ch/supplementary_material/RK_CVPR09/. Acesso em 22 de Novembro de 2017.