

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE
MÉTODOS DE CORREÇÃO DE ILUMINAÇÃO
PARA IMAGENS DIGITAIS**

TRABALHO DE GRADUAÇÃO

Jean Carlo Albiero Berni

**Santa Maria, RS, Brasil
2008**

DESENVOLVIMENTO E IMPLEMENTAÇÃO DE MÉTODOS DE CORREÇÃO DE ILUMINAÇÃO PARA IMAGENS DIGITAIS

por

Jean Carlo Albiero Berni

Trabalho de Graduação apresentado ao Curso de Ciência da
Computação da Universidade Federal de Santa Maria (UFSM, RS),
como requisito parcial para a obtenção do grau de
Bacharel em Ciência da Computação

Orientador: Prof. Dr. José Antônio Trindade Borges da Costa

Trabalho de Graduação N° 248

Santa Maria, RS, Brasil

2008

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**DESENVOLVIMENTO E IMPLEMENTAÇÃO DE
MÉTODOS DE CORREÇÃO DE ILUMINAÇÃO
PARA IMAGENS DIGITAIS**

elaborado por
Jean Carlo Albiero Berni

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA:

Prof. Dr. José Antônio Trindade Borges da Costa
(Presidente/Orientador)

Prof. Dr. Marcos Cordeiro d'Ornellas

Prof. Dr. Cesar Tadeu Pozzer

Santa Maria, 29 de janeiro de 2008.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

DESENVOLVIMENTO E IMPLEMENTAÇÃO DE MÉTODOS DE CORREÇÃO DE ILUMINAÇÃO PARA IMAGENS DIGITAIS

Autor: Jean Carlo Albiero Berni

Orientador: Prof. Dr. José Antônio Trindade Borges da Costa

Local e data da defesa: Santa Maria, 29 de janeiro de 2008.

A correção da iluminação irregular ou *vignetting* é um dos primeiros passos para o processamento de imagens digitais de microscopia ótica e eletrônica, sendo determinante para a qualidade dos resultados obtidos por métodos automáticos de análise de quantitativa. Neste trabalho foram implementados dois métodos, o desfoque gaussiano e a interpolação bicúbica, para extrair o padrão de iluminação de imagens das quais uma imagem de fundo ou *background* não foi adquirida separadamente. Como ambos são métodos assistidos pelo usuário, que deve fornecer amostras da imagem e alguns parâmetros, uma interface gráfica foi desenvolvida na qual o usuário realiza estas operações interagindo diretamente com a imagem, tornando intuitivo o processo de extração do *background*. Uma vez conhecido o padrão de iluminação, pode-se corrigir a imagem original por meio de uma subtração ou divisão. Os métodos foram implementados utilizando a linguagem *Java* e as bibliotecas *JAI* (Java Advanced Imaging) e *ImageIO* e foram integrados ao sistema de processamento e análise de imagens Arthemis, desenvolvido pelo Laboratório de Computação Aplicada - LaCA/UFSM.

Palavras-chave: correção de iluminação; microscopia ótica; iluminação irregular; desfoque Gaussiano, interpolação bicúbica.

ABSTRACT

Trabalho de Graduação
Graduation in Computer Science
Universidade Federal de Santa Maria

DEVELOPMENT AND IMPLEMENTATION OF VIGNETTING CORRECTION METHODS FOR DIGITAL IMAGES

Author: Jean Carlo Albiero Berni
Advisor: Prof. Dr. José Antônio Trindade Borges da Costa

The vignetting correction is one of first steps for the digital processing of images obtained from light and electron microscopy, being a determining factor for the quality of the results achieved by automatic methods of quantitative analysis. In this work two methods were implemented, namely the Gaussian smoothing and the bicubic interpolation, to extract the illumination pattern of images from which a background was not separately acquired. Since both methods are assisted by the user, who must input some parameters and image samples, a graphical interface was developed where the user performs these operations by directly interacting with the images, so that the process of background extraction becomes intuitive. Once the illumination pattern is known, the original image can be corrected by means of a subtraction or a division. The methods were implemented using the *Java* language and the *JAI* (Java Advanced Imaging) and *ImageIO* libraries and were integrated to the image processing and analysis system Arthemis, developed by Laboratório de Computação Aplicada - LaCA/UFSM.

Keywords: vignetting correction; optic microscopy; irregular background; Gaussian Smoothing; Bicubic Interpolation.

LISTA DE FIGURAS

Figura 1: Caminho ótico percorrido pelos raios luminosos dentro de um microscópio de luz transmitida.	10
Figura 2: Exemplos de imagens com iluminação irregular.	11
Figura 3: Seleção do tamanho da máscara por comparação com uma região representativa da textura da cena.	17
Figura 4: Imagem com iluminação irregular e os padrões de iluminação extraídos por aplicações sucessivas do filtro gaussiano. Uma, duas e três passagens.	17
Figura 5: Seleção dos pixels que representam o fundo de imagem.	18
Figura 6: Aceitabilidade de um sistema (Adaptado de Rocha, 2003, p.18).	24
Figura 7: Formato de acesso transversal disponibilizado pela JAI.	30
Figura 8: Cópia de bordas na operação de convolução.	33
Figura 9: Modelagem UML dos métodos de correção.	35
Figura 10: Modelagem UML dos métodos de correção por subtração e divisão.	36
Figura 11: Modelagem UML para geração do background por desfoque Gaussiano.	37
Figura 12: Modelagem UML para geração do background por interpolação bicúbica.	37
Figura 13: Janela principal da aplicação.	39
Figura 14: Interface do método desfoque Gaussiano.	40
Figura 15: Interface do método interpolação bicúbica.	41
Figura 16: Interface principal para o método de correção de iluminação no Arthemis.	42
Figura 17: Interface do método desfoque Gaussiano no Arthemis.	43
Figura 18: Interface do método interpolação bicúbica no Arthemis.	44
Figura 19: Interface para correção da iluminação irregular em lote.	45
Figura 20: Correção de <i>vignetting</i> por subtração do padrão de iluminação gerado por desfoque Gaussiano. (a) Imagem original. (b) <i>Background</i> . (d) Imagem corrigida.	47
Figura 21: Comparação entre os histogramas da região central (linha preta) e do canto inferior direito (linha azul) da imagem da Figura 20. (a) Imagem original. (b) Imagem corrigida por desfoque Gaussiano. (c) Imagem corrigida por interpolação bicúbica.	48

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
GUI	Graphical User Interface
IHC	Interação Humano-Computador
JAI	Java Advanced Imaging
LaCA	Laboratório de Computação Aplicada
RGB	Red Green Blue
ROI	Region Of Interest
UML	Unified Modeling Language
XML	Extensible Markup Language

SUMÁRIO

1. INTRODUÇÃO.....	10
2. MÉTODOS DE CORREÇÃO DE ILUMINAÇÃO.....	14
2.1. Métodos de extração do padrão de iluminação	15
2.1.1. Desfoque Gaussiano	15
2.1.2. Interpolação bicúbica.....	17
2.2. Correção da imagem original	19
2.2.1 Operações aritméticas entre imagens	19
2.2.2 Operações aritméticas implementadas	21
2.3. Combinações entre os métodos de correção.....	21
2.4. Contribuição para a usabilidade dos métodos	22
2.4.1. A interação humano-computador (IHC).....	23
2.4.2. Aceitabilidade em sistemas de interação humano-computador.....	24
2.4.3. Usabilidade em sistemas de interação humano-computador.....	25
2.4.4. A interface do método de desfoque Gaussiano	26
2.4.5. Contribuições para a interface do método interpolação bicúbica.....	26
3. APRESENTAÇÃO E IMPLEMENTAÇÃO DA FERRAMENTA.....	27
3.1. Arquitetura.....	27
3.2. Arthemis	28
3.3. Estendendo a biblioteca JAI	28
3.3.1. Escrevendo novos métodos	29
3.3.2. A classe CorrectSub	31
3.3.3. A classe CorrectDiv	31
3.3.4. As classes CorrectSubDescriptor e CorrectDivDescriptor.....	32
3.3.5. A classe GenGauss	32
3.3.6. A classe GenInterp	33
3.4. Modelagem dos métodos de correção	34
3.4.1. Métodos de correção por subtração e divisão.....	36
3.4.2. Método de geração do background por desfoque Gaussiano	36
3.4.3. Método de geração do background por interpolação bicúbica.....	37
3.5. Interface independente.....	38
3.5.1. Interface principal.....	38
3.5.2. Interface para o desfoque Gaussiano	39
3.5.3. Interface para a interpolação bicúbica	40
3.6. Integração ao Arthemis.....	41
3.6.1. Interface principal no Arthemis.....	41
3.6.2. Interface para o desfoque Gaussiano	42
3.6.3. Interface para a interpolação bicúbica	43
3.7. Sistema de processamento em lote	44

4. RESULTADOS	47
5. CONCLUSÃO.....	49
REFERÊNCIAS	50
ANEXO A – Artigo publicado no SIBGRAPI 2007	52

1. INTRODUÇÃO

Nas últimas décadas tem ocorrido uma crescente utilização de imagens digitais nas mais diversas áreas do conhecimento (ISO, 1997). Como arquivo digital, a imagem pode ser visualizada, armazenada, copiada, compartilhada, transmitida e distribuída com rapidez e sem perda de informação.

Sistemas onde câmeras digitais são acopladas a microscópios óticos tornaram-se padrão nas áreas de aplicação da microscopia quantitativa. Essa configuração conquistou espaço favorecida pelos baixos custos das câmeras digitais. Entretanto, devido às características do caminho ótico percorrido pela luz, desde a fonte de iluminação até a amostra e depois da amostra até a câmera (ver **Figura 1**), a luminosidade da cena fotografada exibirá uma variação mesmo para objetos que possuem a mesma refletância.

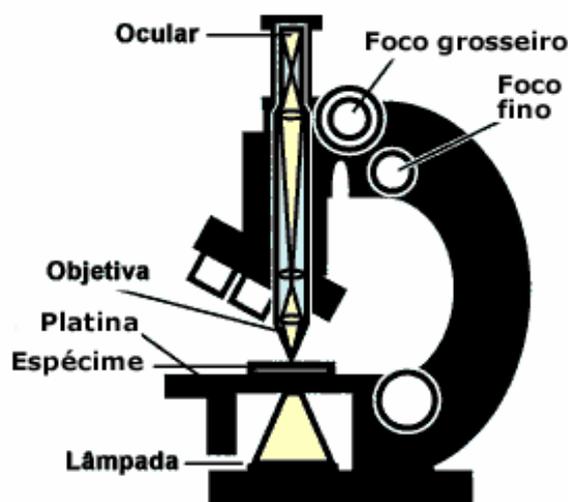


Figura 1: Caminho ótico percorrido pelos raios luminosos dentro de um microscópio de luz transmitida.

Exemplos deste efeito são mostrados na **Figura 2**, nas quais se nota uma maior luminosidade na parte central, que vai diminuindo até as bordas da imagem capturada. Este padrão de iluminação é conhecido como *vignetting* e embora seja muitas vezes atribuído à distribuição irregular da luz sobre a imagem fotografada (INOUE, 1997), também ocorre para cenas em que a iluminação é perfeitamente homogênea.

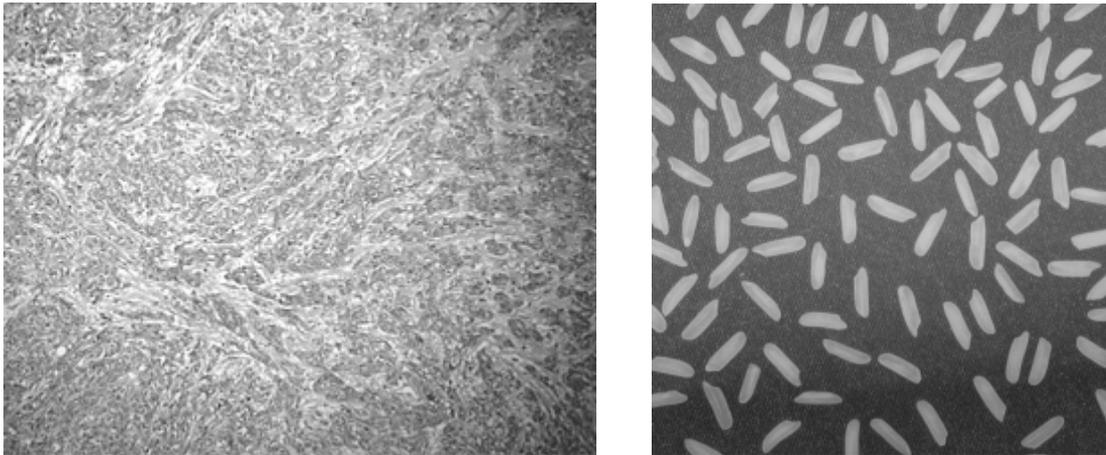


Figura 2: Exemplos de imagens com iluminação irregular.

O efeito de sombreamento das bordas também ocorre em imagens de microscopia eletrônica, ainda que a origem da irregularidade no brilho tenha outras causas. Neste caso, a cena ou campo a ser imageado não é iluminado, mas varrido por um feixe de elétrons que incide com energia de alguns keV. Os elétrons transmitidos, retroespalhados ou secundários são detectados, formando imagens cujos contrastes estão relacionados à topografia da superfície ou à sua composição química. Devido ao caminho percorrido pelos elétrons incidentes e detectados, também neste caso um objeto perfeitamente homogêneo formará uma imagem com luminosidade irregular. Assim, uma vez que, em ambos os casos, os efeitos produzidos são os mesmos, os mesmos métodos de correção de iluminação podem ser utilizados tanto para imagens adquiridas ao microscópio ótico como ao microscópio eletrônico.

A correção da iluminação irregular é necessária não somente por razões estéticas. Ela é, quase sempre, um precursor indispensável para a análise da imagem, uma vez que a medida do brilho é freqüentemente usada como parâmetro nas operações de processamento de imagens (RUSS, 2000). Assim, se não for corrigido, este defeito de aquisição afeta todas as demais etapas de processamento e análise da imagem, incluindo a segmentação e a extração de suas características (WANG, 2005). De fato, para alguns métodos de análise, e em

particular para a análise linear de histogramas (BORGES DA COSTA, 2003; MIRANDA, 2004), é indispensável que a mesma intensidade corresponda sempre ao mesmo objeto. Ao aplicar estes métodos de análise, a irregularidade da iluminação pode fazer com que objetos de mesma natureza sejam classificados como se fossem de naturezas diferentes, apenas por estarem distribuídos em posições diferentes sobre o campo imageado, comprometendo assim a sua quantificação.

Para a correção do *vignetting* existem dois tipos de situação: 1) o padrão de iluminação, ou *background*, é conhecido e registrado separadamente da imagem do campo de interesse para ser eliminado posteriormente, ou 2) o *background* não é conhecido e precisa ser extraído da própria imagem antes de ser eliminado. Uma vez de posse do padrão de iluminação, pode-se operar sobre a imagem original eliminando o defeito por meio de uma subtração ou divisão entre a imagem e seu *background*.

Neste trabalho foram implementados dois métodos de extração de *background* para correção de iluminação irregular através de uma interface que foi desenvolvida para que o usuário opere diretamente sobre as imagens com defeito, fornecendo de modo intuitivo e transparente os dados e parâmetros de entrada. A ferramenta desenvolvida permite ainda a utilização de padrões de iluminação previamente adquiridos e a realização de correção em *batch*, sobre lotes de imagens adquiridas sob as mesmas condições de iluminação, como ocorre em exames de rotina nos laboratórios de análises clínicas, ou de controle de qualidade de indústrias. Neste caso após extrair o padrão de iluminação para um determinado lote de imagens, é desejável que todo o lote seja processado sem a intervenção repetitiva do usuário, aumentando assim a produtividade.

Os métodos implementados para a extração do padrão de iluminação da imagem, quando este não é conhecido, foram o desfoque Gaussiano e a interpolação bicúbica. No primeiro aplica-se um filtro passa-baixa com distribuição Gaussiana que elimina os detalhes texturais e evidencia o padrão de iluminação subjacente (RUSS, 2000). No segundo, colhem-se amostras de um fundo homogêneo às quais é ajustado um polinômio que é usado para gerar, por interpolação, o padrão de iluminação em porções da imagem cobertas por objetos diversos (LEONG et al, 2003). Estes métodos foram escolhidos por permitirem tratar a maioria das imagens de microscopia analisadas no Laboratório de Computação Aplicada - LaCA/UFSM, as quais estão relacionadas a trabalhos de pesquisa nas áreas de metalurgia, ciência dos materiais, mineralogia, petrologia e patologia, realizadas em colaboração com outros grupos de pesquisa na UFSM e em outras instituições.

Ambos os métodos de extração de *background* implementados são bem conhecidos e amplamente descritos na literatura de processamento de imagens (INOUE, 1997; RUSS, 2000; LEONG, 2003; WANG, 2005). A contribuição deste trabalho consiste em facilitar o uso destes métodos e aumentar a sua exatidão com o desenvolvimento de ferramentas que auxiliam o usuário na escolha do tamanho do filtro usado no primeiro método e na seleção de amostras do *background* no segundo. Assim, a interface desenvolvida permite que o usuário abstraia o valor numérico do tamanho do filtro Gaussiano, dando um significado visual para este parâmetro quando o usuário compara visualmente uma máscara de tamanho selecionado com a textura da imagem. Já a seleção das amostras usadas na interpolação bicúbica é guiada por uma grade que garante uma distribuição homogênea de amostras sobre a imagem.

O sistema foi implementado através da linguagem de programação Java e as bibliotecas para processamento de imagens *JAI* e *ImageIO*. A ferramenta produzida foi incorporada ao sistema de processamento e análise de imagens Arthemis, que é desenvolvido pelo LaCA/UFSM. Do mesmo modo que outros grandes sistemas de softwares voltados à análise e processamento de imagens que são desenvolvidos em camadas (VÖLTER, 2002), os métodos discutidos neste projeto são concebidos e foram integrados ao *framework* Arthemis utilizando o padrão arquitetural *Layer*, descrito por Buschmann (BUSCHMANN, 1996).

No Arthemis estão disponíveis diversas ferramentas para processamento e análise de imagens, e a sua organização arquitetural define que cada método seja considerado um *plugin* e independente dos outros. Os métodos só operam sobre as imagens abertas no *framework*. Essas características permitem ao Arthemis uma maior flexibilidade na adição ou remoção de métodos de processamento e análise de imagens. Os métodos desenvolvidos nesse trabalho são independentes do Arthemis, ou seja, somente a camada de interface se comunica com o *framework*. Essa abordagem permite que os métodos possuam uma interface independente ou sejam integrados a qualquer outro *framework* apenas com modificações na camada mais externa da aplicação.

Este trabalho está organizado em cinco capítulos, cujo conteúdo é descrito sucintamente a seguir. No Capítulo 2 são apresentados os métodos de extração de *background* e de correção de iluminação. O Capítulo 3 trata da implementação da ferramenta, da arquitetura do sistema e das interfaces de interação com o usuário. No Capítulo 4 são apresentados os resultados atingidos e, por fim, o Capítulo 5 apresenta as conclusões e sugestões para trabalhos futuros.

2. MÉTODOS DE CORREÇÃO DE ILUMINAÇÃO

Existem câmeras digitais dedicadas exclusivamente a aquisição de imagens em microscópios óticos que implementam a função *white shading correction* em seus programas de aquisição. Essa função consiste em armazenar fotos da lâmina do microscópio sem amostra e depois efetuar uma divisão ou subtração sobre as próximas imagens obtidas das amostras. Para a aplicação deste método é preciso obter fotos da lâmina vazia a cada mudança de objetiva ou de filtros óticos do microscópio, das lentes redutoras do dispositivo adaptador da câmera ao microscópio (*mount*), da resolução da câmera, ou das condições de iluminação do campo, isto é, a cada alteração nas condições de aquisição.

Quando o procedimento de correção de iluminação não é feito no momento da aquisição, seja porque o sistema de aquisição não conta com este recurso, ou porque ele não é usado ou, ainda, quando não se observam as alterações mencionadas acima antes de proceder a correção, obtêm-se imagens nas quais o padrão de iluminação não é conhecido. Nessas circunstâncias, recriar as condições de aquisição para fotografar um campo homogêneo onde a imagem do padrão de iluminação projeta-se é quase sempre inviável. Assim, se faz necessário frequentemente recorrer a outras soluções para a recuperação do *background*.

Os métodos de extração do padrão de iluminação implementados neste trabalho podem ser aplicados tanto a imagens em tons de cinza, como a imagens coloridas, multi-espectrais, ou multi-modais. Para isso, as mesmas operações são realizadas separadamente em cada canal gerando o *background* correspondente e, por fim, a imagem é reconstituída.

A implementação dos métodos de correção de iluminação depende da disponibilidade de um conjunto de operações básicas de processamento de imagens, que incluem a aplicação de filtros (ver Seção 2.2.1.), seleção de regiões de interesse (*ROIs*) (ver página 19), e operações aritméticas entre imagens, as quais são descritas a seguir.

2.1. Métodos de extração do padrão de iluminação

Antes de aplicar a operação aritmética que vai compensar a irregularidade de iluminação, é preciso conhecer o padrão de iluminação ou *background*.

Temos então dois tipos de situação: 1) o padrão de iluminação é conhecido e registrado separadamente da imagem do objeto de interesse, ou 2) o padrão de iluminação não é conhecido e precisa ser extraído da própria imagem antes de corrigir a imagem original. Quando o padrão de iluminação é conhecido aplicam-se diretamente as operações aritméticas definidas na Seção 2.2 entre a imagem original e o *background*, resultando na imagem corrigida.

O principal interesse deste trabalho foi o desenvolvimento de ferramentas eficientes para a implementação de soluções para a segunda situação, onde o *background* precisa ser extraído da imagem que se pretende corrigir. Abaixo são descritos os dois métodos para a extração do *background*: o desfoque Gaussiano e a interpolação bicúbica de amostras representativas do fundo da imagem.

2.1.1. Desfoque Gaussiano

Este método aplica-se aos casos em que a cena adquirida é constituída por uma textura homogênea, cujos elementos são bem menores do que a cena completa. Neste caso o padrão de iluminação pode ser tratado como um ruído de baixa frequência, que pode ser extraído por um filtro passa-baixa (LEONG, 2003). O objetivo deste método é produzir um “embaçamento” da imagem original eliminando os detalhes da textura e evidenciando as diferenças de iluminação.

O desfoque gaussiano é produzido pela aplicação sucessiva de um filtro cujos elementos representam uma distribuição gaussiana, dada pela equação

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right], \quad (1)$$

onde (x, y) representam as coordenadas dos pixels cobertos pela máscara e σ é o desvio padrão. A aplicação deste método exige, portanto, a entrada de dois parâmetros pelo usuário, a saber, o desvio padrão, σ , e o número de aplicações do filtro. Os valores destes parâmetros que produzem os melhores resultados dependem fortemente da textura e da resolução com a

qual a imagem foi adquirida. Assim, o usuário precisa testar diferentes valores, o que pode ser um processo bastante demorado e cansativo, quando feito às cegas, isto é, fornecendo os valores numéricos e verificando o resultado obtido.

Uma maneira de tornar este processo mais simples e eficiente consiste em calcular o valor de σ com base no tamanho, L , da máscara, expresso em número de pixels. No presente trabalho adotamos a escolha usual, $\sigma = (L-1)/4$. Quanto maior for a máscara, maior será o desvio padrão e também maior será a perda de foco ou “embaçamento” produzido na imagem pela aplicação do filtro.

Na prática, o tamanho da máscara que produz os melhores resultados é aproximadamente igual ao tamanho dos elementos de textura da imagem. Máscaras muito menores produzem padrões de iluminação que ainda contém detalhes dos objetos imageados, enquanto máscaras muito maiores não capturam as variações locais de iluminação. Com o auxílio desta informação propusemos um método para selecionar o tamanho adequado da máscara gaussiana, que consiste em selecionar um elemento da imagem que seja representativo da cena completa, isto é, um elemento que, se fosse reproduzido regularmente, resultaria em uma imagem com o mesmo aspecto da imagem original, conforme ilustra a **Figura 3**. Deste modo, o usuário abstrai a definição explícita do desvio padrão ou do tamanho da máscara de distribuição Gaussiana guiando-se apenas pelo aspecto da imagem.

Na interface gráfica que desenvolvemos, após o tamanho da máscara ser definido, o usuário pode testar o número de convoluções necessário para produzir um padrão de iluminação suave.

Na **Figura 4** é mostrada uma imagem com iluminação irregular e os padrões de iluminação extraídos por desfoque gaussiano com tamanho e número de convoluções variáveis. Uma, duas e três passagens. O padrão de iluminação da direita é o que melhor corresponde ao que se observa no fundo da imagem original.

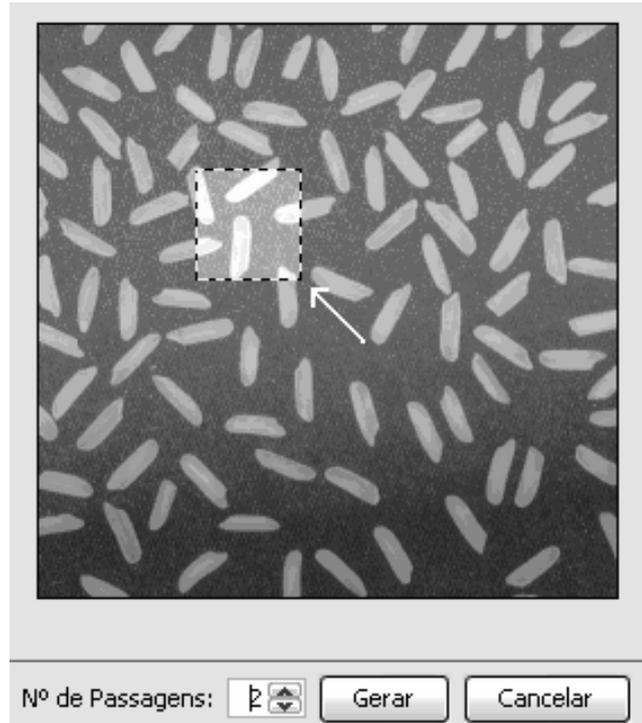


Figura 3: Seleção do tamanho da máscara por comparação com uma região representativa da textura da cena.

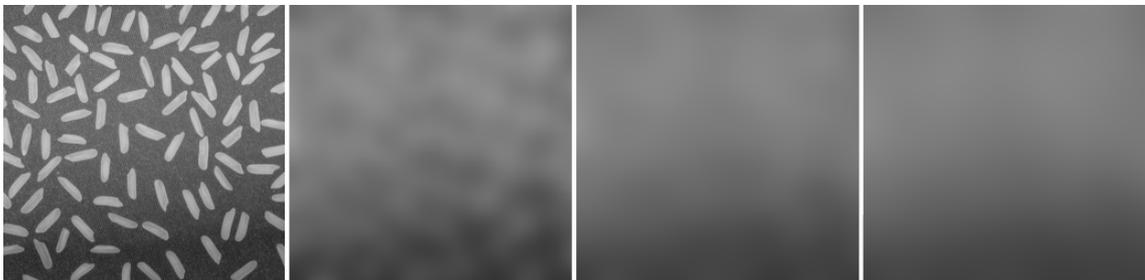


Figura 4: Imagem com iluminação irregular e os padrões de iluminação extraídos por aplicações sucessivas do filtro gaussiano. Uma, duas e três passagens.

2.1.2. Interpolação bicúbica

O método de interpolação aplica-se aos casos em que um campo homogêneo, ou fundo, se destaca dos objetos em primeiro plano. A sua implementação exige que o usuário forneça amostras representativas da luminosidade do fundo em pontos distribuídos regularmente sobre toda a cena. Aos pontos selecionados pelo usuário é ajustado um polinômio bicúbico nas coordenadas (x, y) , de acordo com a equação

$$B(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} ix^{i-1} y^j, \quad (2)$$

onde a_{ij} são os parâmetros a serem ajustados. Esse polinômio é então usado para interpolar todos os demais pontos de uma imagem do mesmo tamanho da imagem a ser corrigida. Assim, através deste polinômio será possível reconstruir o padrão de iluminação da imagem para posterior correção da imagem original.

A qualidade do padrão de iluminação gerado por este método depende do número e da regularidade da distribuição dos pontos selecionados. Por isso, na interface gráfica implementada, uma grade retangular é sobreposta à imagem de modo a guiar o usuário na seleção de amostras representativas do fundo da imagem, conforme ilustra a **Figura 5**. A fim de eliminar ruído de alta frequência, o tom de cinza médio de cada *ROI* é calculado. A matriz composta com os valores dos tons de cinza obtidos deste modo em cada elemento da grade é tomada como uma imagem de baixa resolução do padrão de iluminação. Esta imagem é então ampliada por interpolação bicúbica até o tamanho da imagem original.



Figura 5: Seleção dos pixels que representam o fundo de imagem.

O número de elementos da grade de seleção dos pontos representativos do fundo pode ser ajustado, aumentando ou diminuindo o número de linhas e colunas. Para cada retângulo

deve ser criada uma máscara de seleção dos pixels do fundo, sendo posicionada pelo usuário. Quando mais a imagem for subdividida, maior será a amostragem de pixels do fundo e, conseqüentemente, melhor será o resultado da interpolação, gerando um padrão de iluminação mais próximo daquele da imagem original.

2.2. Correção da imagem original

Após ser gerado o padrão de iluminação podemos operar sobre a imagem original com a finalidade de corrigi-la. Duas operações foram implementadas para este fim, a saber, a subtração e a divisão (RUSS, 2000).

2.2.1 Operações aritméticas entre imagens

As operações aritméticas entre imagens funcionam de forma semelhante a operações aritméticas entre números naturais. Os valores das intensidades de brilho de cada pixel, em cada canal da imagem, são utilizados para o processamento da operação. Assim, por exemplo, na soma de duas imagens as intensidades dos pixels correspondentes às coordenadas (x, y) de cada imagem são somados e o resultado é armazenado em uma terceira imagem, nas mesmas coordenadas.

Entretanto, uma vez que a faixa de valores possíveis dos níveis de intensidade é limitada, ocorre muitas vezes que o resultado da operação entre as intensidades das duas imagens de entrada ultrapasse o limite superior ou fique abaixo do limite inferior desta faixa. Assim, é preciso decidir que valor atribuir ao pixel correspondente na imagem resultado sempre que uma destas situações ocorrer. A escolha mais simples consiste em simplesmente cortar o valor resultante da operação no limite superior ou inferior da escala de intensidades, isto é, atribuir o valor máximo da escala sempre que o resultado da operação ultrapassar este limite e o valor mínimo sempre que o resultado ficar abaixo deste outro limite. Neste caso, a imagem resultante apresentará regiões de saturação, em que todos os contrastes são eliminados o que pode ser um efeito indesejado para muitas aplicações.

De fato, a forma de tratar excessões nas operações aritméticas entre imagens, depende no tipo de aplicação pretendida. Tomando novamente o exemplo da soma, uma possibilidade é realizar a média aritmética dos valores de intensidade das imagens de entrada. Essa é a escolha feita quando se pretende reduzir ruído de alta frequência utilizando várias imagens de

um mesmo campo sob as mesmas condições de aquisição. Uma outra possibilidade é a realização de uma média ponderada, quando o objetivo é produzir uma superposição de imagens ou uma marca d'água.

Como se vê, embora as operações entre imagens recebam os nomes das operações aritméticas, elas são de fatos redefinidas a fim de produzir os melhores resultados para a aplicação pretendida.

As operações aritméticas usualmente utilizadas em processamento de imagens são: “adição”, “subtração”, “multiplicação” e “divisão”. As operações “subtração” e “divisão” usadas na implementação dos métodos de correção de iluminação são detalhadas e comentadas a seguir.

Subtração

A subtração é frequentemente utilizada como um meio de ressaltar a diferença entre duas imagens. Neste caso, toma-se o valor absoluto da diferença entre os valores de intensidade das imagens de entrada, de modo que regiões claras na imagem de saída correspondem a locais onde as duas imagens de entrada diferem muito, e regiões escuras correspondem a locais onde elas diferem pouco. Para o nivelamento de imagens por subtração de *background* (RUSS, 2000), deseja-se preservar a relação entre as intensidades da imagem a ser corrigida. Para isso, acrescenta-se um valor fixo de intensidade a todos os pixels da imagem resultado, conforme a definição apresentada na Seção 2.2.2.

Divisão

A operação aritmética de divisão, que consiste em dividir pixel a pixel o valor da intensidade de uma imagem pela outra, é utilizada para a remoção de *background* quando sistemas de aquisição de imagens com resposta linear (ver Seção 2.3) são utilizados (RUSS, 2000). Neste caso a correção é feita pela divisão da imagem original pelo padrão de iluminação ou *background*. O resultado é um mapa de refletância, com valores compreendidos entre 0 e 1, que deve ser convertido para a escala de intensidades correspondente ao tipo de imagem processada.

Pelas razões apresentadas na Seção 2.3, no presente trabalho, o que chamamos de “divisão” é, na verdade, uma “multiplicação” pelo “negativo” da imagem do *background*. A “multiplicação” envolve uma divisão do resultado do produto das duas imagens de entrada

pelo valor máximo da escala de intensidades, a fim de limitar o resultado à faixa de valores da escala de intensidades. O “negativo” de uma imagem, por sua vez, é obtido quando as intensidades são invertidas em relação ao centro da escala. Assim, o valor máximo de intensidade é convertido no mínimo, e vice-versa, o segundo maior valor é convertido no segundo menor valor, e vice-versa, e assim sucessivamente até cobrir todos os valores de intensidade.

2.2.2 Operações aritméticas implementadas

Neste trabalho, a subtração do padrão de iluminação $B(x, y)$ da imagem de entrada $I(x, y)$ é realizada segundo equação

$$O(x, y) = I(x, y) - B(x, y) + C, \quad (3)$$

onde $O(x, y)$ é a imagem de saída e C é o *offset* usado para ajustar o brilho da imagem resultante, o qual é escolhido de modo que o menor valor de $O(x, y)$ seja *zero*, preservando a relação entre as intensidades dos objetos presentes na imagem original.

A operação de “divisão” entre as imagens é realizada neste trabalho pela equação

$$O(x, y) = \frac{I(x, y) \cdot [255 - B(x, y)]}{255} + C. \quad (4)$$

onde a operação $[255 - B(x, y)]$ corresponde a tomar o negativo da imagem do padrão de iluminação, conforme descrito na Seção anterior, e C é o *offset* usado para ajustar o brilho da imagem resultante. O valor 255 corresponde ao máximo da escala de intensidade em imagens de 8 bits ou em canais de cor de imagens coloridas de 24 bits, as quais constituem a maioria dos casos usualmente encontrados.

2.3. Combinações entre os métodos de correção

Para a correção do defeito de *vignetting* podemos combinar os métodos estudados neste trabalho de formas distintas. Para a geração do padrão de iluminações podem ser usados o

desfoque Gaussiano ou interpolação bicúbica e para efetivar a correção da imagem defeituosa, podem ser usadas as operações de subtração e divisão.

O usuário do sistema de correção deve observar certas características das imagens que serão corrigidas, as quais permitem identificar qual método irá produzir o resultado mais satisfatório, a saber, o tamanho dos objetos de interesse na imagem, a disposição dos mesmos e o contraste destes com o fundo.

O método de extração de *background* por desfoque Gaussiano produz melhores resultados quando a imagem a ser corrigida é constituída por uma textura homogênea, onde os elementos de interesse da imagem são bem menores do que a imagem completa. Já o método de geração do padrão de iluminação por interpolação bicúbica produz melhores resultados quando aplicado a imagens em que o fundo se destaca em relação à amostra, ou seja, existe um contraste muito grande entre os objetos de interesse e seu o *background*. Este é também o método recomendado quando os objetos de interesse forem relativamente grandes em relação à imagem e estiverem homogeneamente distribuídos. Neste caso, ainda que não haja um fundo homogêneo, o método pode ser aplicado se houverem objetos de mesma refletância distribuídos homogeneamente sobre o campo, os quais serão usados no processo de amostragem descrito anteriormente.

Depois de extraído o padrão de iluminação, deve-se escolher a operação a ser usada para a remoção do *vignetting*. Esta não é uma escolha simples. Em princípio seria necessário conhecer o tipo de resposta à intensidade luminosa exibido pelo sistema de aquisição. A divisão aritmética deve ser usada quando o sistema de aquisição exibe uma resposta linear, isto é, a o nível de intensidade registrado na imagem varia na mesma taxa que a intensidade do sinal incidente sobre o sistema de aquisição. Em outras palavras, se o dispositivo de aquisição for uma câmera fotográfica, a resposta será dita linear se o brilho registrado na imagem dobrar quando a intensidade da luz incidente dobrar. Como raramente esse é o caso e em geral a forma da resposta é bastante complexa ou não totalmente conhecida, a operação realizada entre a imagem original e o *background* pode variar dependendo do sistema de aquisição. A subtração, conforme definida na Eq. (3) produz resultados satisfatórios em um grande numero de casos. Já a operação de “divisão”, definida na Eq. (4) é uma alternativa que testamos com sucesso para várias imagens.

2.4. Contribuição para a usabilidade dos métodos

A principal contribuição deste trabalho é facilitar e melhorar o desempenho no uso dos métodos de correção de iluminação através do desenvolvimento de uma interface entre o usuário e os métodos de correção. Segundo Wood (WOOD, 1999), as interfaces devem prover um meio de interação com a aplicação de uma forma intuitiva e natural. Portanto, o simples fato dos métodos de correção funcionarem de forma satisfatória não justifica sua utilidade se não permitir com que os operadores da aplicação consigam utilizá-los com eficiência.

2.4.1. A interação humano-computador (IHC)

A relação entre humanos e computadores é uma área que se preocupa com o projeto, implementação e avaliação de sistemas computacionais interativos bem como o estudo dos fenômenos essenciais que envolvem a comunicação desses sistemas com seres humanos (ROCHA, 2003).

A evolução da performance humana na utilização de computadores e de sistemas de informação é reflexo da crescente pesquisa em interfaces ao longo das últimas décadas. As medições de qualidade e eficiência têm sido possíveis através da utilização de ferramentas computacionais para análise de dados e de contribuições das áreas da Psicologia Social e Organizacional, da Psicologia Cognitiva, de Fatores Humanos ou Ergonomia, da Sociologia, da Linguística, do Design, da Filosofia, da Engenharia, da Antropologia e Inteligência Artificial.

Interfaces humano-computador vêm alterando a rotina de muitas pessoas como, por exemplo, médicos, que utilizam sistemas computacionais para auxiliar diagnósticos precisos, professores e alunos, em ambientes de ensino e aprendizagem, e em diversas atividades que se utilizam de computadores e equipamentos de alta-tecnologia. À medida que esse nível de interação ganha força, também é cada vez mais comum encontrarmos sistemas que provocam medo e frustração em seus usuários ao se depararem com interfaces complexas e de difícil entendimento.

O estudo da IHC carece de informação sobre o comportamento humano noções do funcionamento da tecnologia e, ainda, as formas como ambos se relacionam precisa apresentar-se de forma objetiva. O seu principal objetivo é produzir sistemas que possuam interfaces usáveis, seguras e funcionais, bem como melhorar a utilidade, usabilidade e efetividade de sistemas computacionais. Até pouco tempo atrás acreditava-se que interfaces bem documentadas seriam suficientes para que os usuários tivessem aproveitamento

satisfatório na realização de tarefas. Ou seja, o software não oferecia condições para que o usuário compreendesse de forma lógica o que estava sendo proposto.

Dessa forma, se faz necessário o emprego de técnicas que propiciem um melhor entendimento por parte do operador do programa e, assim, se garanta a obtenção dos resultados esperados com melhor aproveitamento dos recursos disponíveis.

2.4.2. Aceitabilidade em sistemas de interação humano-computador

Os objetivos da Interface Homem-Computador (IHC) são melhor englobados através da aceitabilidade do sistema (NIELSEN, 1995). A aceitabilidade dos sistemas computacionais está diretamente relacionada à combinação das aceitabilidades prática e social conforme representa o esquema mostrado na **Figura 6**.

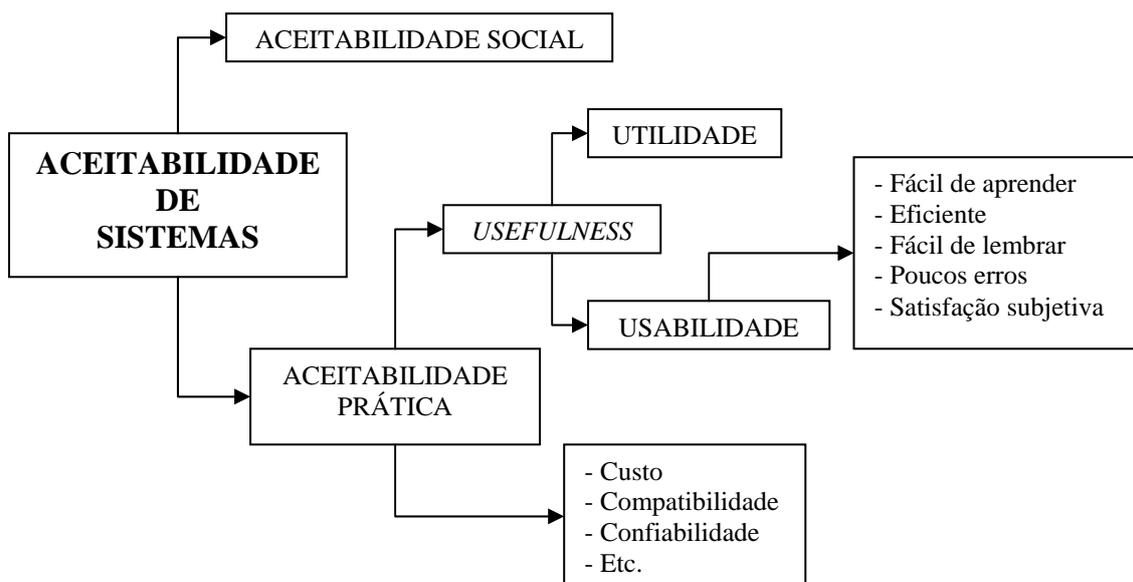


Figura 6: Aceitabilidade de um sistema (Adaptado de Rocha, 2003, p.18).

A aceitabilidade social diz respeito à aversão à tecnologia que modificações no ambiente empresarial podem causar nos funcionários. Afinal, a falta de qualificação ou conhecimento do novo ambiente pode fazer com que as pessoas sintam medo de perder seus empregos por não compreenderem o processo funcional do novo sistema.

A aceitabilidade prática refere-se a custos, compatibilidade, confiabilidade, utilidade e usabilidade do sistema. Na sua avaliação verificam-se características que permitem classificar o sistema como de fácil compreensão, e se este possui uma disposição dos elementos da

interface que seja de fácil acesso e memorização. Dentro da aceitabilidade prática encontramos o conceito de *Usefulness* que avalia a eficiência proporcionada pela sua utilização, ou seja, a sua usabilidade.

A qualificação da usabilidade do sistema, embora leve em consideração detalhes subjetivos e que variam de usuário para usuário, ajuda de forma significativa na definição dos parâmetros importantes que devem ser apresentados ao operador do software. Na próxima seção são apresentadas algumas características desejáveis que devem ser contempladas no desenvolvimento de uma interface de software.

2.4.3. Usabilidade em sistemas de interação humano-computador

Usabilidade é definida como a medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação num contexto específico de uso (ISO, 1998). A usabilidade pode ser expressa como a combinação entre o usuário, o sistema e a tarefa que deve ser executada. Para ser considerada usável uma interface deve apresentar características que satisfaçam as necessidades impostas pela tarefa efetuada pelo usuário. Os fatores que expressam a usabilidade são:

- **Facilidade de aprendizado:** permite ao usuário novato compreender inicialmente como utilizar e alcançar, pela experiência, um nível mais elevado de eficiência e eficácia na utilização do sistema.
- **Flexibilidade da interação:** agrupa os aspectos que permitem uma grande variação de maneiras pelas quais o usuário pode trocar informações com o sistema.
- **Robustez da interação:** define os aspectos de interação que suportam a realização e a avaliação dos objetivos que são estipulados na realização das tarefas.

Em vista disso, procurou-se levar em consideração aspectos delineados nas seções anteriores para o desenho das interfaces dos métodos de correção de iluminação. De fato, uma das principais colaborações no desenvolvimento dos métodos propostos neste trabalho diz respeito a interface com o usuário. Nas próximas duas Seções os detalhes das interfaces para o software de correção são ilustrados.

2.4.4. A interface do método de desfoque Gaussiano

No método para geração do padrão de iluminação por desfoque Gaussiano, quando o usuário faz uma seleção sobre a imagem, como mostrado na **Figura 3**, ele está obtendo de forma transparente alguns parâmetros necessários para o cálculo do *background*. Como base na seleção é repassado ao método o tamanho da máscara de convolução que será aplicada na imagem. Este parâmetro determina o desvio padrão e, pela equação de Gauss, os valores que preencherão a máscara para convolução.

Ao se guiar somente pelo aspecto da imagem são garantidos facilidade de aprendizado e flexibilidade de interação. Em termos práticos o aprendizado se torna simples, pois o operador sempre tomará por base características intrínsecas à imagem e a capacidade de reconhecimento de padrões permitirá ao operador aplicar as mesmas definições às imagens que são semelhantes.

O principal objetivo da interface para o método de desfoque Gaussiano é permitir que usuário utilize somente de critérios baseados nas características da imagem para definir os parâmetros do método, não sendo necessário o conhecimento aprofundado da parte teórica por trás da operação.

2.4.5. Contribuições para a interface do método interpolação bicúbica

Para a geração do *background* por interpolação bicúbica o nível de interação entre o método e o seu operador é maior do que o método Gaussiano. O usuário precisa selecionar amostras representativas do fundo ao longo de toda a imagem como é mostrado na **Figura 5**. Embora a dificuldade de identificar onde as seleções devem ser feitas seja menor, o nível de exigência e demora na demarcação das áreas representativas do *background* são bem maiores.

Como é exigido pelo método que sejam amostrados pontos sobre toda a imagem, a grade serve de guia para o operador, indicando onde este deve efetuar pelo menos uma seleção. O objetivo dessa interface é guiar o usuário, para que este selecione, de forma regular, pontos distribuídos por toda imagem. A facilidade de aprendizado e robustez da interface são contemplados de forma satisfatória.

3. APRESENTAÇÃO E IMPLEMENTAÇÃO DA FERRAMENTA

3.1. Arquitetura

Os métodos de correção de iluminação discutidos neste trabalho serão incorporados ao sistema de processamento e análise de imagens Arthemis. Este sistema possui várias ferramentas agregadas para pré-processamento, segmentação, e extração de características.

O desenvolvimento do Arthemis segue o padrão arquitetural de camadas descrito por Buschmann, no qual o usuário faz uma requisição à camada superior e essa requisição é tratada pelas camadas inferiores (BUSCHMANN, 1996). O resultado obtido em cada camada é retornado, camada a camada, até ser apresentado ao usuário. Assim, o sistema de correção de iluminação apresenta uma camada central ou núcleo, uma camada de aplicação e uma de apresentação. O principal objetivo dessa arquitetura é tornar o software flexível em relação às suas funcionalidades, decompondo-o em blocos de funções similares ao seu nível de abstração (BUSCHMANN, 1996).

Consistentemente com a arquitetura do Arthemis, o sistema de correção de iluminação foi estruturado em três camadas: a interface gráfica (primeira camada), os métodos (segunda camada) e núcleo (terceira camada). A interface (*GUI*) gráfica possibilita a interação direta dos usuários com o sistema. Na camada intermediária encontram-se os métodos de geração do padrão de iluminação e de correção das imagens. Nela são recebidas as imagens e os parâmetros de operação. A partir dessa camada é acessada a camada núcleo onde, finalmente, as operações sobre os pixels são efetivadas utilizando a biblioteca *JAI* e os métodos estendidos como nativos da biblioteca.

3.2. Arthemis

O sistema de processamento e análise de imagem Arthemis é desenvolvido no Laboratório de Computação Avançada – LaCA/UFSM e conta com a colaboração de alunos de graduação e mestrado que desenvolvem seus trabalhos de pesquisa. Através do Arthemis toda a produção tecno-científica do grupo é centralizada em uma ferramenta que possui as características básicas para o processamento e análise de imagens, como por exemplo, funções de abertura e salvamento de imagens, zoom, rotação, acesso a banco de imagens entre outros.

O Arthemis foi estruturado com o objetivo de receber contribuições na forma de *plugins*. Com uma interface completamente configurável através de arquivos *XML*, permite a fácil integração dos novos métodos que vão sendo desenvolvidos bem como a internacionalização do *framework*, com traduções para os principais idiomas.

A interface configurável do Arthemis permite que este seja adaptado conforme o destino da aplicação, embora os métodos de processamento e análise de imagens sejam praticamente os mesmos para todas as áreas como patologia, mineralogia, sensoriamento remoto, etc., a terminologia empregada deve corresponder ao vocabulário familiar a cada profissional em cada especialidade. A interface do *framework* permite a inclusão ou remoção de módulos que sejam de interesse de determinada área de aplicação, ou seja, o Arthemis é moldado conforme as necessidades do usuário, garantindo a flexibilidade e facilidade de utilização em cada ambiente específico.

3.3. Estendendo a biblioteca JAI

Embora a API JAI suporte um grande número de operação sobre imagens, ela foi projetada desde o início para encorajar programadores a escreverem extensões da biblioteca ao invés de manipularem as imagens diretamente. A JAI virtualmente permite que qualquer algoritmo de processamento de imagens possa ser adicionado a sua API e ser usado como se fosse uma parte nativa da biblioteca (JAI Guide, 1999).

O mecanismo para adicionar funcionalidades a API possui múltiplos níveis de encapsulamento e complexidade. Isso permite aos programadores adicionar operações simples à biblioteca bem como operações complexas e com vários níveis de processamento.

Neste trabalho os métodos básicos da JAI foram estendidos para a criação dos métodos de correção da terceira camada de forma que pudessem ser chamados como nativos da biblioteca, facilitando e padronizando a estrutura da aplicação.

3.3.1. Escrevendo novos métodos

Para ampliar a API JAI através da criação de novos operadores é necessário estender algumas classes com operações básicas para o processamento de imagens da biblioteca. Após, deve-se utilizar o mecanismo de registro da JAI para permitir com que os novos métodos possam ser invocados de forma nativa pelo programador.

As classes necessárias para a criação de novos métodos são:

- Uma classe estendendo a classe *OpImage* ou alguma de suas subclasses. Essa classe é responsável pelo real processamento da imagem, ou seja, as operações pixel a pixel;
- Uma classe estendendo a classe *OperationDescriptor*, que descreve o nome da operação, lista de parâmetros e demais detalhes do método.

As subclasses de *OpImage* que já estão implementadas na JAI são descritas a seguir:

- *AreaOpImage*: classe abstrata para operadores que necessitam computar uma região ao redor de um determinado pixel;
- *PointOpImage*: classe abstrata para operadores que necessitam processar sobre o conjunto de pixels de uma imagem e armazenar os valores correspondentes em uma imagem resultado;
- *SourcelessOpImage*: classe abstrata para processamento de métodos que não contêm imagens;
- *StatisticsOpImage*: classe abstrata que retorna o processamento sobre uma determinada região da imagem na forma de dados estatísticos.
- *WarpOpImage*: classe abstrata para operações geométricas sobre as imagens.

Após estender a classe *OpImage* ou uma de suas subclasses, existem duas funções que devem ser implementadas, *getTile* e *computeRect*. A função *getTile* é responsável por abrir uma região da imagem para leitura e retornar os valores calculados. A função *computeRect* processa o método no nível dos pixels e é chamada pela *getTile*.

Dentro da função *computeRect* e necessária a utilização dos iteradores da JAI que definem a forma como os pixels serão acessados para leitura ou escrita. Neste trabalho utilizou-se a interface para acesso aos pixels *RectIter* que permite o acesso transversal aos *pixels* da imagem como mostrado na **Figura 7**. As posições de cada *pixel* podem ser acessadas através das seguintes funções: *nextLine()*, *jumpLines()*, *nextPixel()*, *jumpPixels()*, *nextBand()*, *setSample* e *setPixel*.

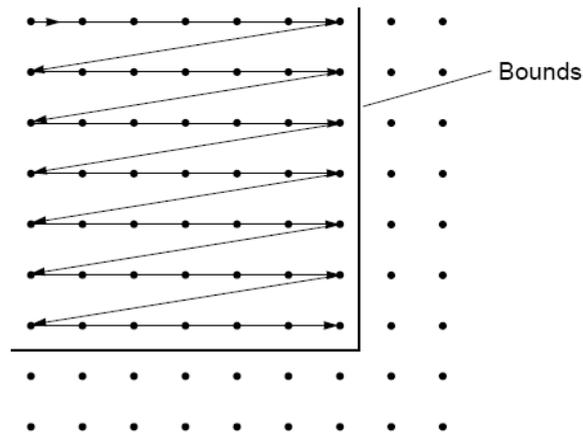


Figura 7: Formato de acesso transversal disponibilizado pela JAI.

Os métodos que são chamados através da JAI pelo comando *JAI.create* precisam estar registrados na biblioteca em *registryFile*, incluído em *jai.core.jar*. Para isso os métodos precisam também estender a interface *OperationDescriptor*, onde serão providas as descrições textuais do método, e definições para o formato das imagens e parâmetros. A inserção destas informações no arquivo de registro da JAI permite com que os novos métodos possam ser acessados como se pertencessem à própria biblioteca.

O tipo abstrato de dado para representação das imagens que são operadas pela biblioteca JAI é o *PlanarImage*. Qualquer imagem que é carregada pela JAI, independente do seu formato, TIFF, BMP, JPEG, etc., responde pelo tipo *PlanarImage*, e implementa os métodos de acesso a seu modelo de cor e pixels conforme o padrão estabelecido pela biblioteca. Assim como as imagens carregadas, os resultados obtidos através das funções da biblioteca também são do tipo *PlanarImage*.

3.3.2. A classe *CorrectSub*

A classe *CorrectSub* corresponde ao método de correção por subtração. Seguindo o padrão da JAI para criação de novos operadores, essa classe estende a classe *PointOpImage* que é a que possui os parâmetros definidos na biblioteca que melhor se adaptam às necessidades do método. A classe *PointOpImage* é utilizada quando se necessita operar sobre todos os pixels e armazenar o resultado obtido uma terceira imagem resultado (ver Seção 3.3.1.). A classe *CorrectSub* recebe como parâmetros para cálculo a imagem original, o seu padrão de iluminação correspondente e o *offset* para o deslocamento de brilho. O cálculo dos valores para cada pixel é aplicado segundo a equação (3) da Seção 2.2.3. O parâmetros são passados ao método através do formato descrito pela biblioteca JAI, que utiliza a classe *ParameterBlock* para sua definição. Então a chamada ao método de subtração pode ser efetuada da seguinte forma:

```
ParameterBlock pb = new ParameterBlock(); // criação da variável pb
que armazena os parâmetros.
pb.addSource(imagemOriginal); // definição de imagem original
pb.addSource(background); // definição do padrão de iluminação.
pb.add(offset); // definição do offset
resultado = JAI.create("CorrectSub", pb); // processamento da
operação através da JAI. Onde os parâmetros são passados via pb.
```

Essa padronização facilita o desenvolvimento de qualquer camada de interface, ao seguir o formato descrito pela JAI qualquer desenvolvedor que conheça a biblioteca pode utilizá-lo de forma coerente com os métodos que já estão implementados na mesma.

3.3.3. A classe *CorrectDiv*

A classe *CorrectDiv* diz respeito ao método de correção por divisão. Da mesma forma que a *CorrectSub* a classe *CorrectDiv* também estende *PointOpImage* por possuir as mesmas características. Os parâmetros para a correção da iluminação também são os mesmos. A diferença consiste na forma de cálculo sobre os pixels, no caso da divisão é aplicada a equação (4) da seção 2.2.3. O parâmetros são passados ao método através da classe *ParameterBlock* e a chamada ao método é feita da seguinte maneira:

```

ParameterBlock pb = new ParameterBlock(); // criação da variável pb
que armazena os parâmetros.
pb.addSource(imagemOriginal); // definição de imagem original
pb.addSource(background); // definição do padrão de iluminação.
pb.add(offset); // definição do offset
resultado = JAI.create("CorrectDiv", pb); // processamento da
operação através da JAI. Onde os parâmetros são passados via pb.

```

3.3.4. As classes *CorrectSubDescriptor* e *CorrectDivDescriptor*

As classes *CorrectSubDescriptor* e *CorrectDivDescriptor* são os descritores das operações apresentadas. Através dos descritores as operações de subtração e divisão serão registradas na JAI, permitindo o acesso nativo por comandos da própria biblioteca. Nas classes estão delineados os detalhes de cada método, permitindo o reconhecimento da quantidade de parâmetros e funções que deverão ser acessadas pelo comando *JAI.create*. Alguns do detalhes definidos nessa classe são:

```

//detalhes do método
private static final String[][] resources = {
    {"GlobalName", "CorrectSub"},
    {"LocalName", "CorrectSub"},
    {"Vendor", "LaCA"},
    {"Description", "Efetua o método de correção de iluminação por
subtração"},
    {"DocURL", ""},
    {"Version", "1.0"}
};

//número de fontes(imagens)
private static final int numSources = 2;

//parâmetros da operação
private static final String[] paramNames = {"offset"};

//classes dos parâmetros.
private static final Class[] paramClasses = { java.lang.Integer.class
};

```

3.3.5. A classe *GenGauss*

Na classe *GenGauss* estão contidos os procedimentos para a execução do método para a geração do padrão de iluminação por desfoque Gaussiano. Os parâmetros que devem ser definidos quando a classe é instanciada são: a imagem original, o número de convoluções da máscara e o tamanho L da máscara. A chamada ao método é feita da seguinte forma:

```
//exemplo de utilização da classe GenGauss
GenGauss gauss = new GenGauss(imagemOriginal, numConv, tamMasc);
resultado = gauss.execute();
```

O resultado é uma *PlanarImage* que representa o padrão de iluminação da imagem utilizada. A função *execute()* é responsável pelas chamadas às funções auxiliares e às operações nativas disponibilizadas pela JAI descritas a seguir.

A variável *tamMasc* será semente para o cálculo dos valores, conforme a equação (1) da seção 2.2.1, que preencherão a máscara para a convolução. Na convolução existe um detalhe importante que deve ser levado em consideração, que diz respeito aos valores utilizados nas bordas da imagem quando a máscara estiver sobre estas. Por exemplo, quando o pixel central da máscara estiver sobre o primeiro pixel da imagem, os pixels da lateral esquerda e superiores da máscara estarão fora da superfície da imagem. Não haveria como calcular o valor para o pixel correspondente na imagem resultado, já que este deve levar em consideração todos os vizinhos. Existem alguns procedimentos usuais em processamento de imagem que minimizam essa falta de informação. Pode-se considerar zero os valores externos à imagem ou definir um valor arbitrariamente. Porém, em ambos os casos, os valores que melhor ajustam-se à imagem não estariam sendo representados com precisão.

Para resolver este problema utilizou-se a técnica de cópia de bordas, que é processada como a ilustração da **Figura 8**.

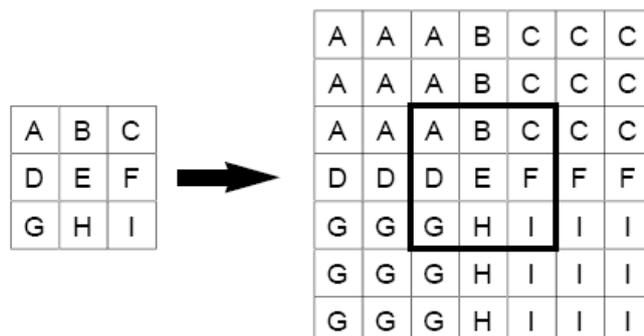


Figura 8: Cópia de bordas na operação de convolução.

3.3.6. A classe *GenInterp*

A classe *GenInterp* possui as funções para operação de geração do *background* por interpolação bicúbica. Os parâmetros que devem ser instanciados na declaração da classe são: a imagem original, o números de linhas e colunas em que a imagem foi dividida pela

interface, e um *ArrayList* das *ROIs* que foram selecionados sobre a imagem. A declaração e execução da classe podem ser feitas da seguinte forma:

```
//exemplo de utilização da classe GenInterp
GenInterp interp = new GenInterp(imagemOriginal, numLin, numColu,
array);
resultado = interp.execute();
```

Do mesmo modo que a classe *GenGauss* o retorno da função *execute()* é uma *PlanarImage* que contém o padrão de iluminação gerado pela interpolação bicúbica. Com base na disposição da grade sobre a imagem, cada *ROI* é identificada levando em consideração sua posição *x, y* de seleção. Para cada elemento da grade sobreposta à imagem são separados as *ROIs* correspondentes, e cada um dos elementos dará origem a um pixel. Assim, é obtida a imagem de baixa resolução que reconstituirá o *background* (ver Seção 2.2.2.). A operação *scale* da JAI combinada com a opção de renderização por interpolação bicúbica é utilizada para a ampliação da imagem de baixa resolução que dará origem ao *background* da imagem.

3.4. Modelagem dos métodos de correção

Seguindo o padrão de desenvolvimento do Arthemis, onde cada método é considerado um *plugin* e deve possuir sua própria interface, os métodos implementados neste trabalho foram modelados para permitir que o acesso seja feito de forma independente. Portanto, qualquer interface de usuário poderá acessar os métodos de correção de forma isolada. As interfaces desenvolvidas devem utilizar os formatos descritos nas quatro seções anteriores para utilização os métodos de correção.

Na **Figura 9** observamos a modelagem *UML* das camadas da aplicação, a primeira camada corresponde a interface com o usuário, representada pela classe *VignettingCorrectioInterface*, que permite o fornecimento dos parâmetros necessários aos métodos de correção. Essa classe acessa as interfaces específicas para cada um dos métodos de geração do *background* e pode ser modelada de acordo com a aplicação que a está utilizando. Através do Arthemis, essa classe é responsável pela obtenção das imagens solicitadas para correção que estão abertas no ambiente. Da mesma forma, se a interface for

independente de *framework*, deve prover uma forma com a qual o usuário indicará as imagens que devem ser processadas. A camada da interface corresponde à parte flexível do sistema.

Na segunda camada encontram-se as classes que comunicam os parâmetros obtidos com o núcleo da aplicação. As classes *CorrectSub*, *CorrectDiv*, *GenGauss* e *GenInterp* descritas nas seções anteriores, são responsáveis pelas chamadas às funções do núcleo, onde o método é processado, e por retornar o resultado obtido para a primeira camada, a interface com o usuário.

A terceira camada disponibiliza os métodos de processamento e análise de imagens que estão implementados na biblioteca JAI e as classes auxiliares com funções úteis à preparação dos parâmetros que serão repassados aos métodos nativos da JAI.

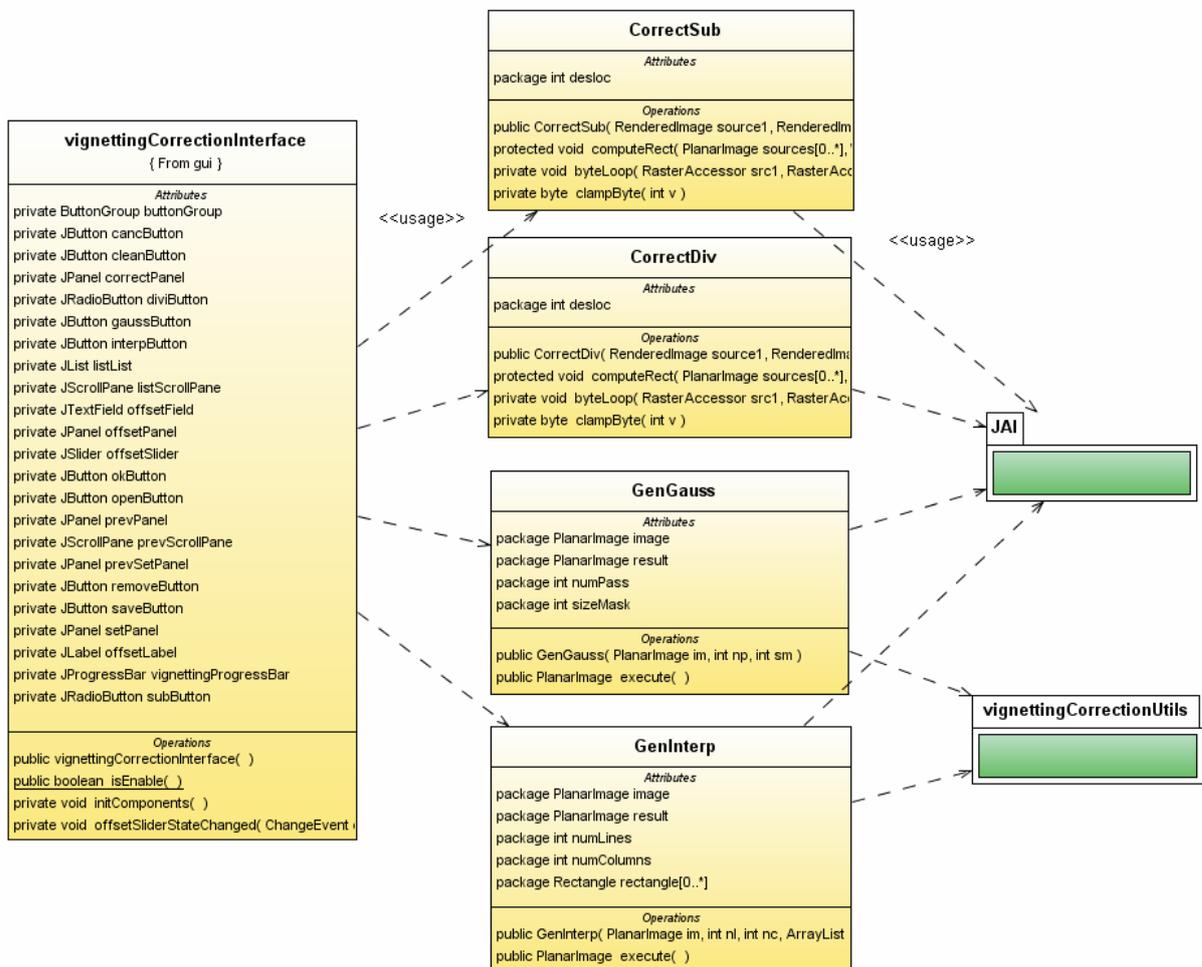


Figura 9: Modelagem UML dos métodos de correção.

A forma como os *plugins* para o Arthemis são desenvolvidos não é padronizada, ficando a cargo de cada desenvolvedor decidir a melhor forma de organizar e modelar o seu método. Dois motivos importantes influenciam a não modelagem dessa parte do sistema, a

quantidade de métodos que já haviam sido desenvolvidos antes da padronização e a flexibilidade para decisões de projeto que melhor se adaptem as necessidades do programador.

3.4.1. Métodos de correção por subtração e divisão

As modelagens para os métodos de correção por subtração e divisão restringem-se à estruturação da interface principal do método. Por se tratarem da última etapa do processamento, ou seja, só serão aplicadas após a definição de todos os parâmetros necessários, as chamadas às classes de correção é direta como mostrado na **Figura 10**.

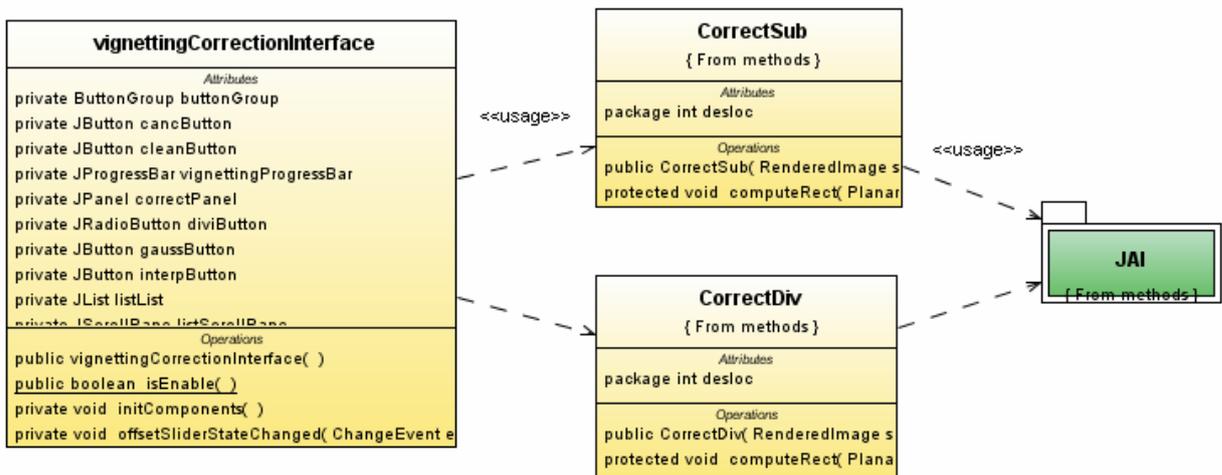


Figura 10: Modelagem UML dos métodos de correção por subtração e divisão.

Fica a cargo da interface verificar se todos os parâmetros já foram definidos antes de acessar os métodos de correção por subtração e divisão.

3.4.2. Método de geração do background por desfoque Gaussiano

Para gerar o padrão de iluminação por desfoque Gaussiano, a interface principal acessa a interface específica do método, representado na **Figura 11** pelo pacote *gaussPanel*. Nesse pacote estão contidas as interfaces que permitem ao usuário definir os parâmetros para o método de desfoque Gaussiano. As interfaces específicas do método não precisam ser modificadas, ao contrário da interface principal, exemplificada como *VignettingCorrectionInterface*, que necessita de adaptação conforme o ambiente utilizado.

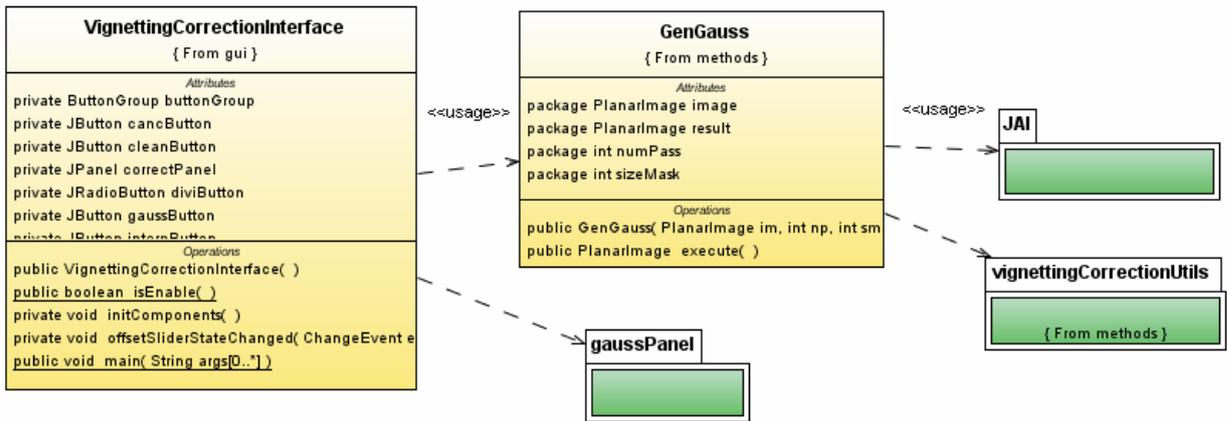


Figura 11: Modelagem UML para geração do background por desfoque Gaussiano.

Assim, a interface principal acessa as classes de interface do pacote *gaussPanel* para que o usuário defina os parâmetros e, após, repassa as definições para a classe *GenGauss* onde se procede a geração do padrão de iluminação.

3.4.3. Método de geração do background por interpolação bicúbica

A modelagem para geração do padrão de iluminação por interpolação bicúbica é semelhante à modelagem para o desfoque Gaussiano. Na interpolação o pacote que é acessado pela interface principal chama-se *interPanel* e nele estão contidos as interfaces específicas para a operação. Na **Figura 12** observamos a modelagem deste método.

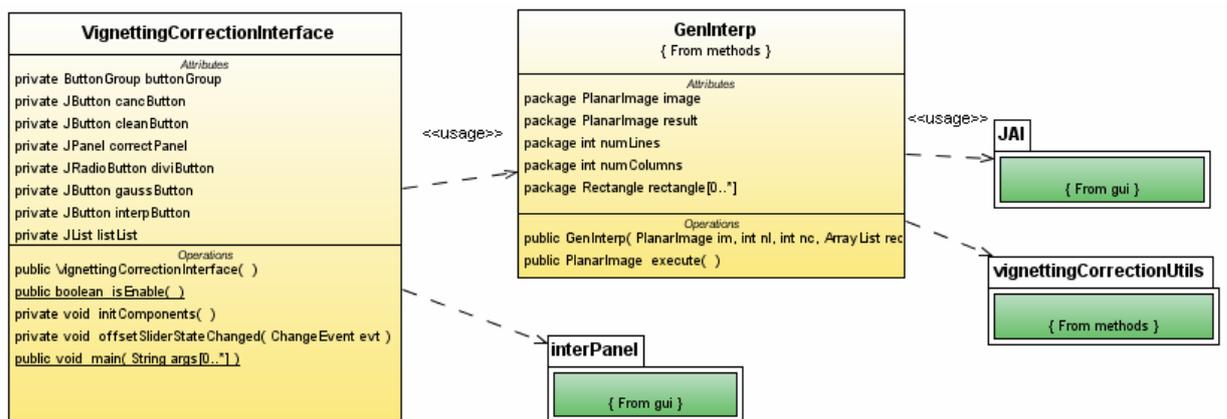


Figura 12: Modelagem UML para geração do background por interpolação bicúbica.

Da mesma forma que o método descrito anteriormente a interface principal deve chamar as classes do pacote *interPanel* e repassar o parâmetros definidos para a função *GenInterp*.

3.5. Interface independente

Para demonstrar a funcionalidade dos métodos e a possibilidade de se modificar facilmente a primeira camada do sistema, desenvolveu-se uma interface independente do *framework* Artemis. Através dela é possível abrir imagens, corrigí-las, salvar padrões de iluminação gerados e os resultados obtidos. Nas próximas seções é discutida sua utilização.

3.5.1. Interface principal

A interface principal possui duas janelas onde são abertas a imagem original e a representação do seu padrão de iluminação. Na **Figura 13** temos a representação da janela principal. O usuário pode abrir uma imagem através de *abrir*, a imagem será mostrada na janela esquerda, intitulada *Imagem Original*. Após a abertura da imagem a ser corrigida, o usuário pode escolher o Método de geração do *background* que melhor se adapta a imagem em questão ou abrir um *background* já existente no canto inferior direito. De posse do padrão de iluminação, situação ilustrada na **Figura 9**, o usuário necessita somente escolher o método de correção, subtração ou divisão, e definir o valor do *offset* para correção de brilho. Por fim, o usuário acessa o botão *Corrigir Iluminação* e a imagem corrigida é mostrada na tela. Existe ainda a opção de salvar o *background* que foi gerado para futuras utilizações.

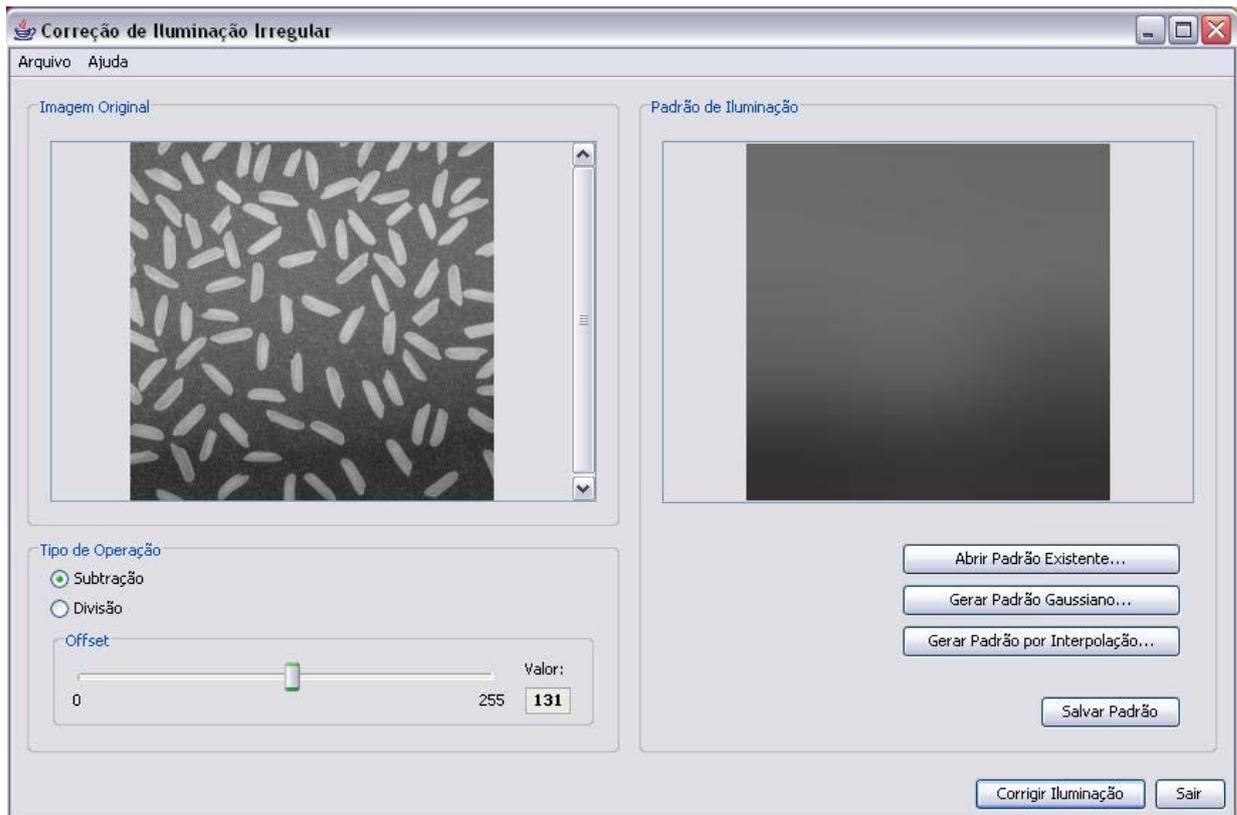


Figura 13: Janela principal da aplicação.

3.5.2. Interface para o desfoque Gaussiano

Para gerar o padrão por iluminação por desfoque Gaussiano utiliza-se interface apresentada na **Figura 14**. Na janela *Gerar Padrão Gaussiano* deve ser definido o número de passagens da máscara de convolução sobre a imagem e uma seleção representativa do menor elemento de textura da imagem, como argumentado na seção 2.2.1.

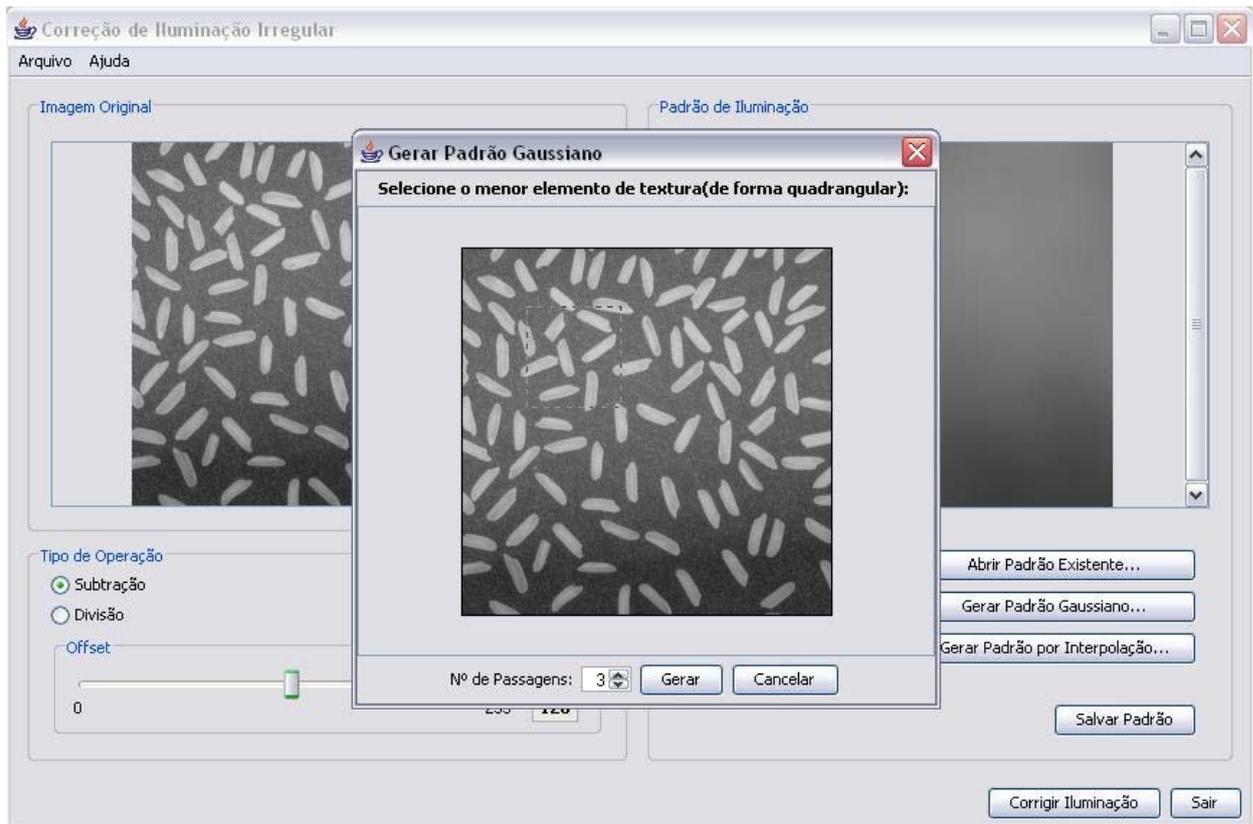


Figura 14: Interface do método desfoque Gaussiano.

3.5.3. Interface para a interpolação bicúbica

Para gerar o padrão de iluminação por interpolação bicúbica encontramos a situação ilustrada pela **Figura 15**. O primeiro passo na interação com o método consiste na divisão de imagem em linhas e colunas, opção encontrada na parte inferior da janela. Após esse procedimento o usuário deve fazer seleções que representam o fundo da imagem em cada quadrante, ou seja, colher amostras do *background* em cada posição da tabela em que a imagem foi dividida.

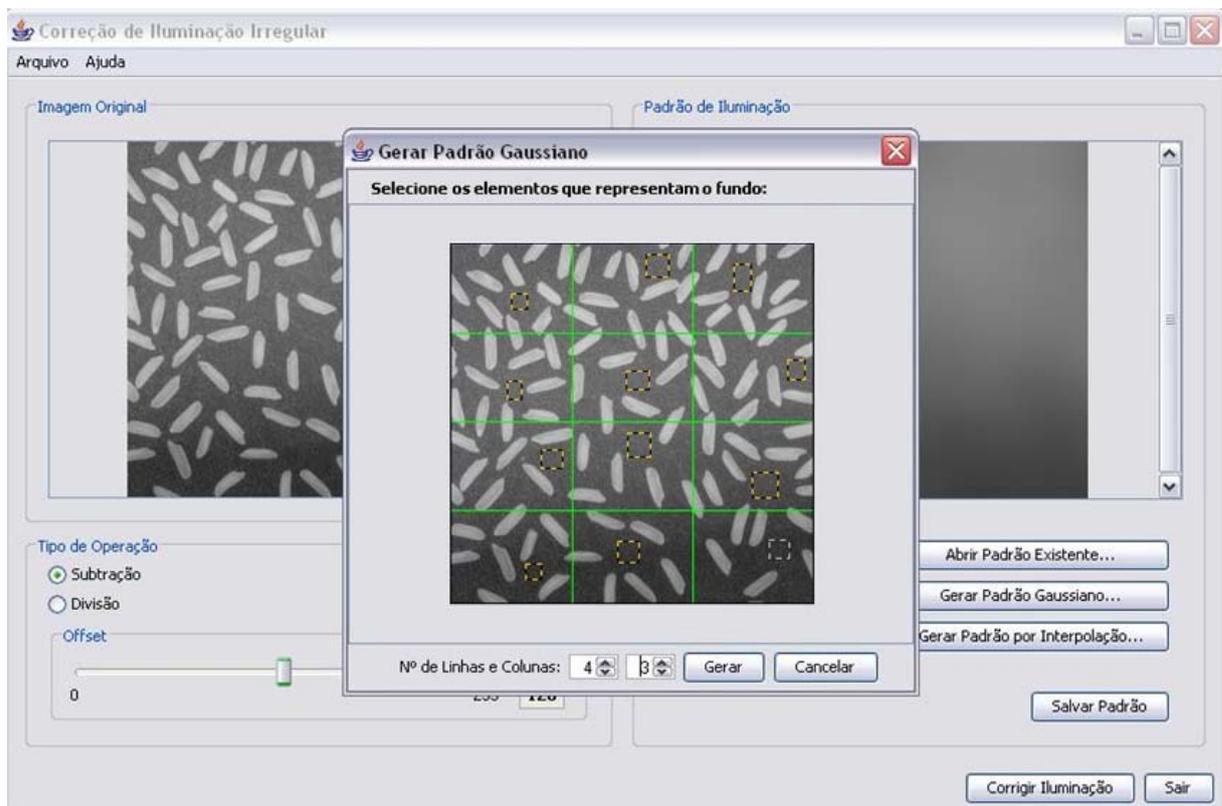


Figura 15: Interface do método interpolação bicúbica.

3.6. Integração ao Arthemis

A extensão do Arthemis é feita através da adição *plugins*, onde cada um deve possuir os métodos para processamento e a interface que será acessada pelo *framework*. O processo para correção de iluminação irregular é semelhante ao descrito nas seções para a interface independente, a diferença básica consiste na forma como a imagem é passada ao método. Na interface independente a imagem é acessada através da abertura de um arquivo, enquanto no Arthemis a imagem já carregada é repassada ao método de correção que retorna o resultado para ser apresentado no *framework*.

3.6.1. Interface principal no Arthemis

A **Figura 16** apresenta a janela de interação para o método de correção. No painel à direita, denominado *Definir Padrão*, é definido o padrão de iluminação que será utilizado na correção. Através do botão *Abrir* pode-se carregar um *background* existente, o botão *Salvar* armazena o padrão selecionado e os botões *Remover* e *Limpar* atuam sobre a lista de padrões

inseridos. Logo abaixo estão os botões *Adicionar Gaussiano +* e *Adicionar por Interpolação +*, que são utilizados para a geração dos *backgrounds*. Conforme os *backgrounds* vão sendo gerados ou abertos, são inseridos na lista. O usuário deve selecionar na lista o melhor resultado calculado para o padrão que será utilizado na correção. Após as definições, cada acesso ao botão *Corrigir* irá inserir uma nova imagem corrigida no *framework*, o usuário pode variar o método de correção, Subtração ou Divisão, e a barra de *Offset*.

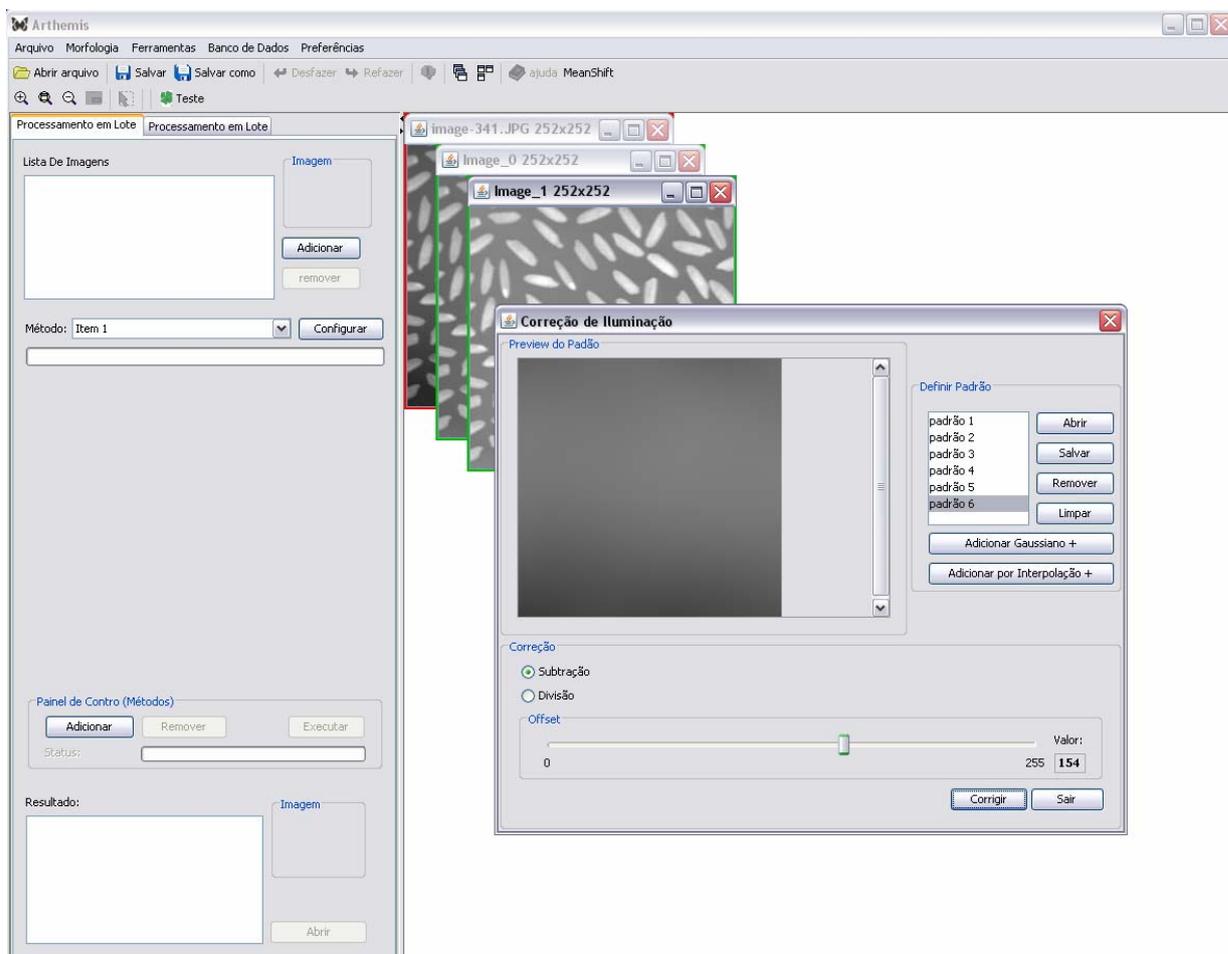


Figura 16: Interface principal para o método de correção de iluminação no Artemis.

3.6.2. Interface para o desfoque Gaussiano

A interface para geração do padrão de iluminação por desfoque Gaussiano funciona da mesma forma como descrito na seção 3.5.2 para a interface independente de *framework*.

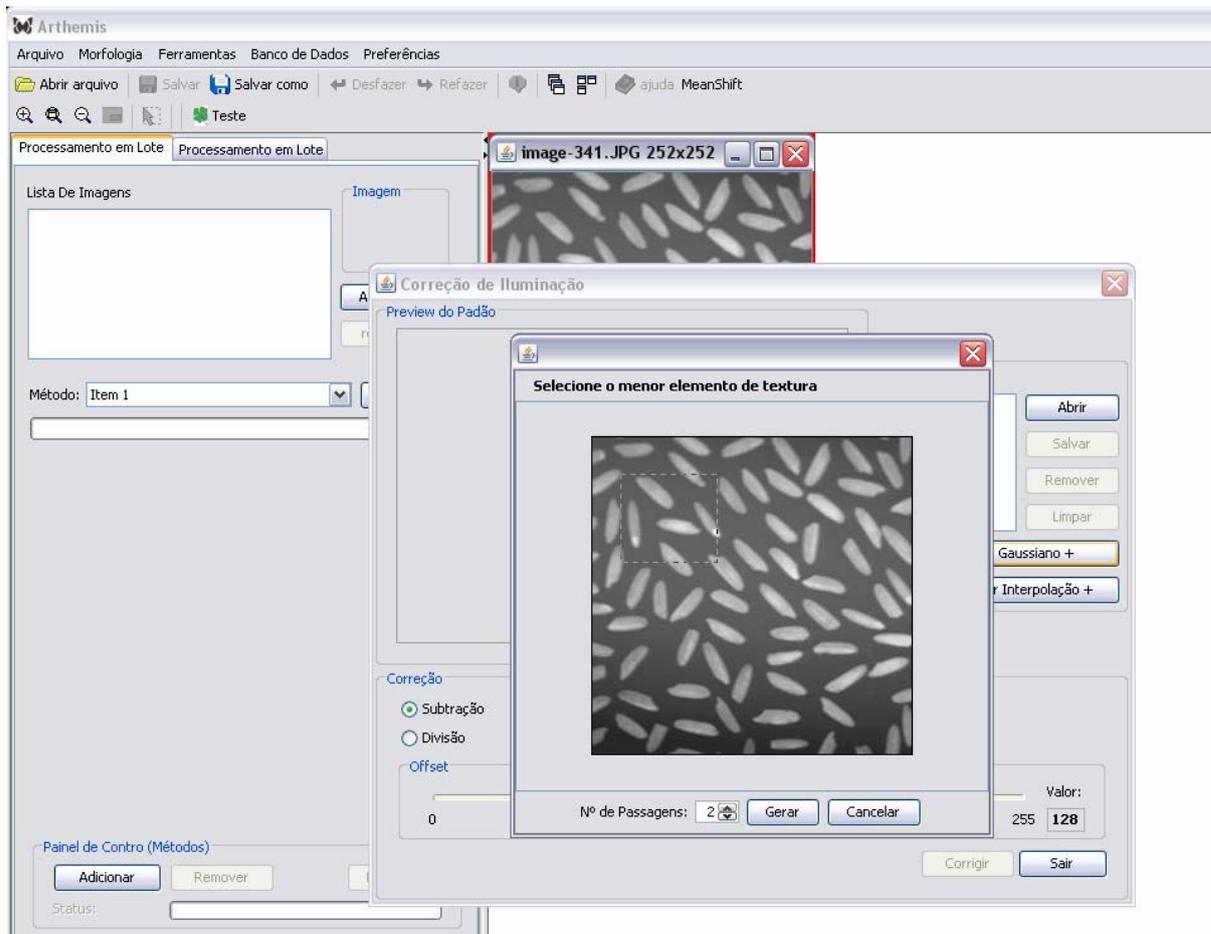


Figura 17: Interface do método desfoque Gaussiano no Artemis.

3.6.3. Interface para a interpolação bicúbica

A **Figura 18** mostra a interface para a geração do padrão de iluminação por interpolação bicúbica dentro do Artemis. A forma de seleção das amostras na imagem é semelhante à descrita na seção 3.5.3. Após o processamento, a imagem resultado é inserida na lista de padrões da interface principal do método.

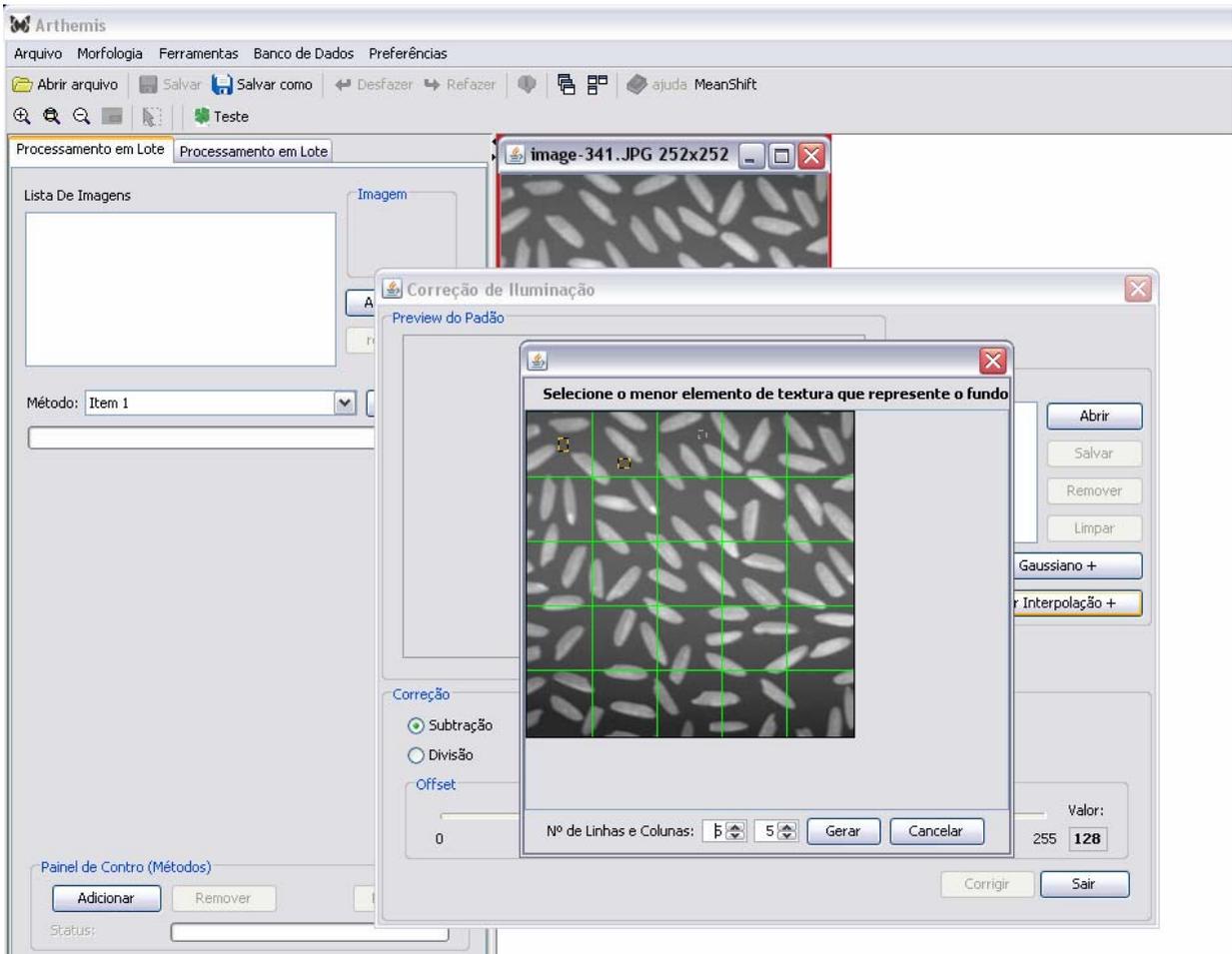


Figura 18: Interface do método interpolação bicúbica no Artemis.

3.7. Sistema de processamento em lote

Uma das principais características apresentadas pelo *framework* Artemis é o suporte para processamento em lote, ou seja, o processamento de grande quantidade de dados sem a intervenção direta do usuário conforme os cálculos vão ocorrendo. Geralmente o processamento e análise de imagens têm utilidade significativa quando aplicados em uma grande quantidade de imagens, proporcionando ao usuário eficiência na obtenção das informações.

Conforme a **Figura 19**, no lado direito da aplicação Artemis encontra-se o painel configurador das operações em lote. Nesse painel é definida a seqüência de operações que podem ser aplicadas em uma lista de imagens abertas pelo usuário. Cada operação intermediária deve ser configurada com seus parâmetros, por exemplo, uma operação de *brilho* precisa ser configurada com o valor desejado antes do lote começar a ser executado. Podemos ter uma seqüência com as seguintes etapas: correção de iluminação e quantificação

das frações de área das imagens. O processamento em lote pode conter o número de etapas que usuário definir. O resultado corresponde ao último método processado pelo lote que leva em consideração todos processamentos definidos na sequência.

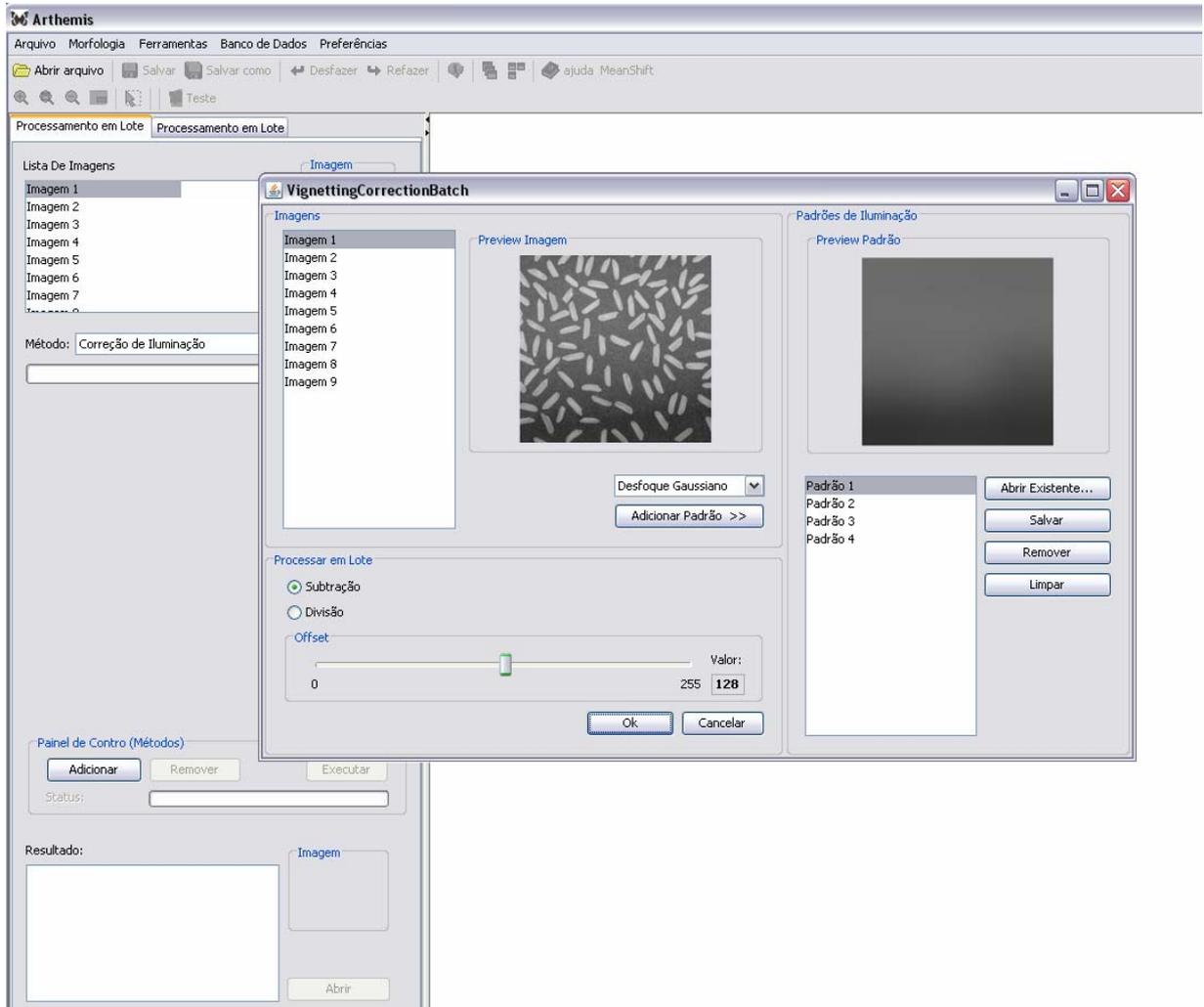


Figura 19: Interface para correção da iluminação irregular em lote.

Ainda na **Figura 19** percebemos a interface para configuração do método de correção de iluminação irregular em lote. No lado direito, dentro do painel *Imagens* encontram-se a lista de imagens do lote, uma janela de *preview* e o botão para adição de padrão de iluminação por desfoque ou interpolação. Quando este botão é pressionado, a janela para configuração do método de geração do *background* é mostrada. Cada padrão gerado é inserido na lista de padrões do lado esquerdo. A lista de padrões pode ser manipulada pelos botões *Abrir Existente*, *Salvar*, *Remover* e *Limpar*. Posteriormente à adição dos padrões, deve ser selecionada o tipo de operação, *Subtração* ou *Divisão*, e o *Offset*. O botão *Ok* confirma as definições para o processamento em lote.

O processamento em lote do método de correção de iluminação calcula o padrão médio de iluminação através de consecutivas médias entre os *backgrounds* da lista. O padrão de iluminação médio final é aplicado sobre todas as imagens do lote utilizando os parâmetros definidos para a correção. Não existe limite para a geração dos padrões de iluminação, e, quanto maior for o número de *backgrounds* melhor será o padrão de iluminação médio.

4. RESULTADOS

Os métodos descritos nas seções anteriores foram implementados e utilizados com sucesso pelos membros do LaCA. Durante os testes, observou-se que tanto o desfoque Gaussiano como a interpolação bicúbica apresentam resultados satisfatórios no que se refere à extração do padrão de iluminação nas situações em que cada um deles se aplica.

Na **Figura 20** é mostrado o resultado da correção de iluminação, onde o padrão de iluminação extraído por desfoque Gaussiano é subtraído da imagem original.

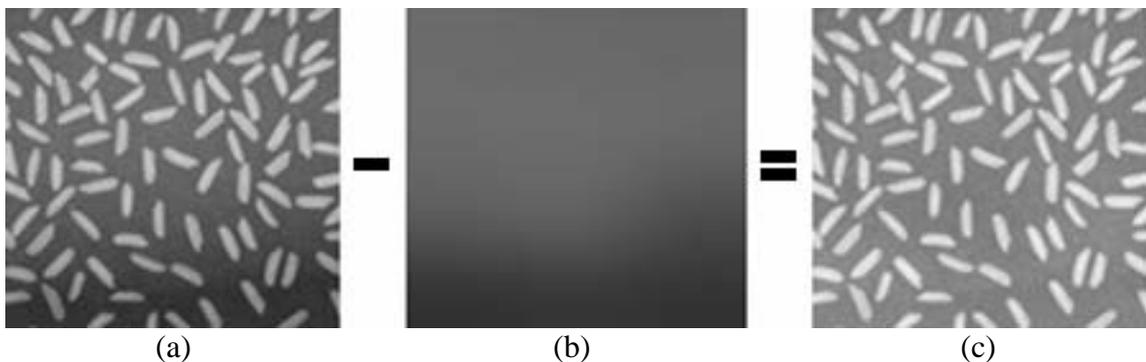


Figura 20: Correção de *vignetting* por subtração do padrão de iluminação gerado por desfoque Gaussiano. (a) Imagem original. (b) *Background*. (d) Imagem corrigida.

A avaliação da qualidade da correção é feita comparando-se os histogramas de objetos de mesma luminosidade em posições diferentes sobre a imagem. No exemplo da **Figura 20**, os objetos de interesse estão dispostos sobre um fundo homogêneo. No gráfico da **Figura 21 (a)**, estão representados os histogramas de amostras do fundo da imagem original na posição central (linha preta), em que aparece mais claro, e no canto inferior direito (linha azul), em que aparece mais escuro. A diferença é bastante evidente.

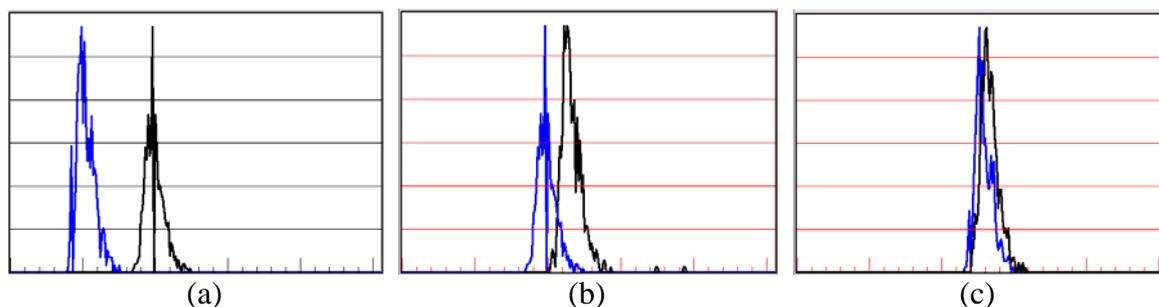


Figura 21: Comparação entre os histogramas da região central (linha preta) e do canto inferior direito (linha azul) da imagem da Figura 20. (a) Imagem original. (b) Imagem corrigida por desfoque Gaussiano. (c) Imagem corrigida por interpolação bicúbica.

Nos gráficos da **Figura 21 (b) e (c)** estão representados os histogramas de amostras do fundo na posição central (linha preta) e no canto inferior direito (linha azul), após as correções por subtração usando os *backgrounds* extraídos respectivamente por desfoque Gaussiano e interpolação bicúbica. Em ambos os casos pode-se observar a redução no contraste. Conforme apresentado, para este caso, o método de extração do *background* que produz a maior redução de contraste, resultando em uma maior coincidência entre os histogramas do fundo em diferentes posições da imagem é o de interpolação bicúbica, conforme esperado, em virtude das características da imagem.

Em comparação com outros sistemas de processamento de imagens, a ferramenta desenvolvida proporcionou um aumento de produtividade pela redução do tempo gasto pelos usuários para selecionar o tamanho do filtro Gaussiano ou fazer a escolha correta dos pontos representativos do fundo.

Os resultados obtidos neste trabalho foram objeto de publicação de um artigo (BERNI, 2007) nos anais SIBGRAPI 2007 Belo Horizonte, MG - Brasil, e apresentação de um pôster no evento.

5. CONCLUSÃO

Neste trabalho, dois métodos foram implementados para a extração do padrão de iluminação irregular em imagens digitais para a sua posterior aplicação na correção das imagens afetadas, a saber, o desfoque Gaussiano e a interpolação bicúbica. Ambos os métodos necessitam de intervenção do usuário, que fornece parâmetros ou seleciona pontos representativos do padrão de iluminação. Para isso foi desenvolvida uma interface gráfica que torna mais fácil e intuitiva a definição destes parâmetros, reduzindo o tempo gasto pelo usuário para fazer a escolha mais adequada dos dados entrada dos métodos.

A ferramenta desenvolvida foi integrada a um sistema de processamento e análise de imagens, o Arthemis, cuja arquitetura facilita a incorporação de novos componentes, melhorando a performance de métodos que se baseiam na medida de intensidades.

A funcionalidade para correção em lote foi adicionada ao sistema com base no padrão definido pelo Arthemis para operações em batch, assim, a correção de conjuntos de imagens adquiridos sob as mesmas condições pode ser processada sem a intervenção direta do usuário em cada etapa do processamento.

Como continuação deste trabalho, novos métodos de correção utilizando o padrão de iluminação extraído serão implementados e testados além do desenvolvimento de um operador de detecção automatizada do melhor resultado que foi gerado. A otimização na avaliação dos resultados será feita através da análise dos histogramas, como foi mostrado no capítulo 4. O melhor resultado é dado pela imagem que tiver diferença mínima entre os histogramas para seleções no *background* ao longo da cena.

REFERÊNCIAS

BERNI, J.; BORGES DA COSTA, J.; D'ORNELLAS M. *Uma interface para métodos de correção de iluminação irregular*. XX Brazilian Symposium on Computer Graphics and Image Processing. Belo Horizonte, MG – Brasil, 2007.

BORGES DA COSTA, J. A. T. *Area fractions by linear analysis of intensity level histograms*. XVI Brazilian Symposium on Computer Graphics and Image Processing. São Carlos, SP – Brasil, 2003.

BUSCHMANN, F.; MEUNIER, R. *Pattern - Oriented Software Architecture: A System of Patterns, volume 1*, John Wiley & Sons, 1996.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION ISO 9241-1: *Ergonomic requirements for office work with visual display terminals (VTDs) – Part 1: General introduction*. Geneva: ISO, 1997.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION ISO 9241-11: *Ergonomic requirements for office work with visual display terminals (VTDs) – Part 11: Guidance on usability*. Geneva: ISO, 1998.

INOUE, S.; SPRING, K. *Video microscopy. The fundamentals, 2nd ed.* New York: Plenum Press, 1997.

LEONG, F.; BRADY, M. and MCGEE, J. *Correction of uneven illumination vignetting in digital microscopy images*. J. Clin Pathol 56, pp. 619-621, 2003.

MIRANDA, A. N. *Quantiphase: Um programa de processamento e análise de imagens para a caracterização da composição e homogeneidade de materiais*. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal de Santa Maria, 2004.

NIELSEN, J. *Usability Inspection Methods*. Conference companion on Human factors in computing systems, 1995.

PRESS, W.; TEUKOLSKY, S.; VETTERLING, W. *Numerical recipes in C++: the art of scientific computing, 3rd*. Ed., Cambridge University Press, 2007.

ROCHA, H. V. da; BARANAUSKAS, M. C. C. *Design e avaliação de interfaces humano-computador*. Campinas: Núcleo de Informação Aplicada à Educação – NIED/UNICAMP, 2003.

RUSS, J. C. *Image Processing Handbook*. CRC Press, Inc., 2000.

SUN DEVELOPER NETWORK (SDN). *Java Advanced Imaging (JAI)*. Disponível em: <[http:// java.sun.com/products/java-media/jai/reference/api/index.html](http://java.sun.com/products/java-media/jai/reference/api/index.html)>.

SUN DEVELOPER NETWORK (SDN). *Java ImageIO Framework*. Disponível em: <http://java.sun.com/products/java-media/jai/downloads/download-iio-1_0_01.html>.

SUN MICROSYSTEMS. *Programming in Java Advanced Imaging. (JAI Guide)*. 1999.

VÖLTER, M.; WOLFF, E. *Server Component Patterns: Component Infrastructures Illustrated with EJB*, 2002.

WANG, M.; PAN, J.; CHEN, S. and LI, H. *A Method of Removing the Uneven Illumination Phenomenon for Optical Remote Sensing Image*. Proceedings of the Geoscience and Remote Sensing Symposium, IEEE International, Volume 5, Issue , 25-29, 2005 pp. 3243–3246, 2005.

WOOD, L. E. *User Interface Design – Bridging the gap from user requirements to design*. Utah: CRC Press LLC, 1999.

ANEXO A – Artigo publicado no SIBGRAPI 2007

Uma interface para métodos de correção de iluminação irregular

Jean Carlo Albiero Berni
José Antônio Trindade Borges da Costa (orientador)
Marcos Cordeiro d’Ornellas (orientador)
GPIM/UFSM – Departamento de Eletrônica e Computação
Universidade Federal de Santa Maria
Centro de Tecnologia/Campus Universitário, Santa Maria, RS – 97105-900
{jeanberni, jatborgesdacosta, marcosdornellas}@gmail.com

Resumo

Este trabalho apresenta uma implementação de dois métodos assistidos pelo usuário para a correção de iluminação irregular em imagens digitais de microscopia ótica e eletrônica, os quais se aplicam a situações em que o padrão de iluminação não é conhecido, sendo necessário gerá-lo a partir da imagem a ser corrigida.. Estes métodos, são acessados através de uma interface gráfica, na qual o usuário colhe amostras do padrão de iluminação e ajusta os parâmetros de entrada dos métodos interagindo diretamente com a imagem, tornando intuitivo o processo de reconstrução do padrão de iluminação.

1. Introdução

A formação de imagens em sistemas óticos é usualmente afetada por distorções e aberrações de natureza variada. Assim, o brilho de objetos cuja superfície é fisicamente homogênea pode apresentar um padrão de variação que está associada, entre outras causas possíveis, à forma e à posição da fonte de luz e ao caminho ótico percorrido pela luz desde o objeto até o dispositivo sensor [1]. O resultado conjunto destes efeitos é um padrão de iluminação irregular que frequentemente necessita ser corrigido, não somente por razões estéticas, mas porque é, quase sempre, um precursor necessário à análise da imagem [2]. Em algumas situações esse padrão é mais claro no centro da imagem, esmaecendo em direção às bordas, sendo então chamado de “vignetting”, nome que passou a ser adotado por alguns autores como sinônimo de todo tipo de irregularidade de iluminação. Este mesmo tipo de defeito de aquisição ocorre também em imagens

de microscopia eletrônica, ainda que a origem da irregularidade no brilho tenha outras causas, uma vez que, neste caso, a imagem não é iluminada, mas varrida por um feixe de elétrons. De qualquer modo, também neste caso um objeto perfeitamente homogêneo formará uma imagem com luminosidade irregular, necessitando de correção.

Duas situações costumam ser encontradas quando se aborda o problema da correção do “vignetting”: 1) o padrão de iluminação é conhecido e registrado separadamente da imagem do objeto de interesse para ser posteriormente usado como máscara de correção, ou 2) o padrão de iluminação não é conhecido e precisa ser extraído da própria imagem a ser corrigida.

Neste trabalho implementamos duas soluções para a extração do padrão de iluminação da imagem, quando este não é conhecido, a saber, o desfoque gaussiano e a interpolação bicúbica. Na primeira aplica-se um filtro passa baixa com distribuição gaussiana que elimina os detalhes texturais e evidencia o padrão de iluminação subjacente. Na segunda, colhem-se amostras de um fundo homogêneo aos quais é ajustado um polinômio que é usado para gerar, por interpolação, o padrão de iluminação em porções da imagem cobertas por objetos diversos [3-5]. O padrão de iluminação gerado é usado para corrigir a imagem original realizando entre estas duas imagens uma operação de subtração ou divisão.

2. Métodos

Os métodos de extração do padrão de iluminação implementados neste trabalho aplicam-se a duas situações bastante frequentes, e embora sejam aqui descritos para imagens em tons de cinza, de oito bits, eles foram de fato implementados para imagens em

cores. Para isso, as mesmas operações são realizadas separadamente nos canais R, G e B.

2.1. Desfoque gaussiano

Este método aplica-se aos casos em que a cena adquirida é constituída por uma textura homogênea, cujos elementos são bem menores do que a cena completa. Neste caso o padrão de iluminação pode ser tratado como um ruído de baixa frequência, que pode ser extraído por um filtro passa-baixa.

O desfoque gaussiano é produzido pela convolução sucessiva de uma máscara cujos elementos representam uma distribuição gaussiana, dada pela equação

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp[-(x^2 + y^2)/2\sigma^2], \quad (1)$$

onde (x, y) representa as coordenadas dos pixels cobertos pela máscara. Em princípio, o desvio padrão, σ , deve ser fornecido pelo usuário. Na prática é mais simples calcular este parâmetro com base no tamanho, L , da máscara. Quanto maior for a máscara, maior será o desvio padrão e também maior será a perda de foco ou “embaçamento” da imagem. No presente trabalho adotamos a escolha usual, $\sigma = (L-1)/4$.

Para selecionar o tamanho adequado da máscara gaussiana, o usuário deve selecionar um elemento da imagem que seja representativa da cena completa, isto é, um elemento que, se fosse reproduzido regularmente, resultaria em uma imagem com o mesmo aspecto da imagem original, conforme ilustra a Figura 1. Deste modo, o usuário abstrai a definição explícita do desvio padrão ou do tamanho da máscara de distribuição Gaussiana guiando-se apenas pelo aspecto da imagem.

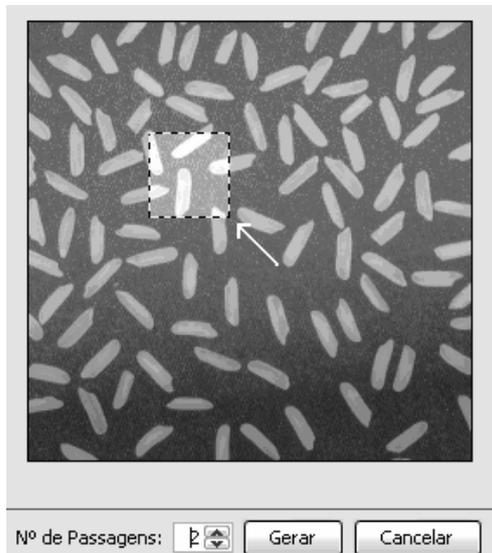


Figura 1: Seleção de região representativa da textura da cena.

Após a máscara ser definida, o usuário pode testar o número de convoluções necessário para produzir um padrão de iluminação suave. Na Figura 2 é mostrada uma imagem com iluminação irregular e os padrões de iluminação extraídos por desfoque gaussiano com tamanho e número de convoluções variáveis. O padrão de iluminação da direita é o que melhor corresponde ao que se observa no fundo da imagem original.



Figura 2: Imagem com iluminação irregular e os padrões de iluminação extraídos por desfoque gaussiano.

2.2. Interpolação bicúbica

Este método aplica-se aos casos em que um campo homogêneo ou fundo se destaca dos objetos em primeiro plano. A sua implementação exige que o usuário forneça amostras representativas da luminosidade do fundo em pontos distribuídos regularmente sobre toda a cena. Aos pontos selecionados pelo usuário é ajustado um polinômio bicúbico nas coordenadas x e y ,

$$B(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} ix^{i-1} y^j, \quad (2)$$

onde a_{ij} representa os parâmetros a serem ajustados. Esse polinômio é então usado para interpolar todos os demais pontos de uma imagem do mesmo tamanho da imagem a ser corrigida.

A qualidade do padrão de iluminação gerado por este método depende do número e da regularidade da distribuição dos pontos selecionados. Por isso, na interface gráfica implementada, uma grade retangular é sobreposta à imagem e, em cada elemento desta grade, o usuário deve posicionar uma pequena máscara de seleção de um conjunto de pixels que melhor representam o fundo da imagem, conforme ilustra a Figura 3. A fim de eliminar ruído de alta frequência, o tom de cinza médio de cada conjunto de pixels selecionado do fundo é calculado. A matriz composta com estes valores é tomada como uma imagem de baixa resolução do padrão de iluminação. Esta imagem é então ampliada por interpolação bicúbica até o tamanho da imagem original.

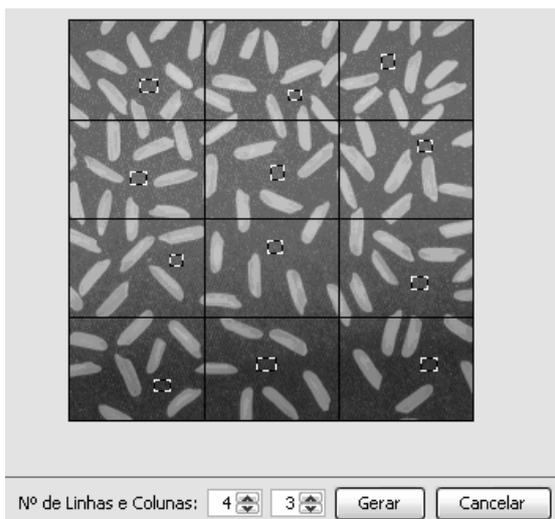


Figura 3: Seleção dos pixels que representam o fundo de imagem.

O número de elementos da grade de seleção dos pontos representativos do fundo pode ser ajustado, aumentando ou diminuindo o número de linhas e colunas. Para cada retângulo é criado automaticamente uma máscara de seleção dos pixels do fundo, a qual deve ser posicionada pelo usuário. Quando mais a imagem for subdividida, maior será a amostragem de pixels do fundo e, conseqüentemente, melhor será o resultado da interpolação, gerando um padrão de iluminação mais próximo daquele da imagem original.

2.3. Correção da imagem original

Após ser gerado o padrão de iluminação podemos operar sobre a imagem original com a finalidade de corrigi-la. Duas operações foram implementadas para este fim, a saber, a subtração e a divisão [2].

A subtração do padrão de iluminação $B(x, y)$ da imagem de entrada $I(x, y)$ é realizada segundo equação

$$O(x, y) = I(x, y) - B(x, y) + C \quad , \quad (3)$$

onde $O(x, y)$ é a imagem de saída e C é o offset usado para ajustar o brilho da imagem resultante.

A operação de divisão entre as imagens é realizada segundo a equação

$$O(x, y) = I(x, y) \cdot [255 - B(x, y)] / 255 + C \quad . \quad (4)$$

De fato, a operação $[255 - B(x, y)]$ corresponde a tomar o negativo da imagem do padrão de iluminação, de modo que o que chamamos de divisão é na realidade uma multiplicação pelo negativo do padrão de iluminação mais um offset C .

Devido às limitações destas operações, novos esquemas de correção devem ser incorporados

como opção ao usuário dependendo da situação encontrada.

3. Implementação

3.1. Bibliotecas

Todo o sistema foi programado na linguagem Java utilizando as bibliotecas JAI (Java Advanced Imaging) e Image IO.

A JAI é uma API para processamento de imagens desenvolvida para a linguagem Java, construída sobre a API Java2D que possui um arcabouço com mais de 100 operações, das quais a grande maioria são nativamente aceleradas [5]. Por ser extensível, possibilita a criação de novos operadores e a sua inclusão na API, de modo que estes passam a ser nativos da biblioteca. Isso torna as chamadas aos novos operadores definidos pelo desenvolvedor exatamente iguais às chamadas aos operadores já disponibilizados juntamente com a JAI.

Assim os métodos para correção de iluminação foram desenvolvidos estendendo métodos básicos nativos e incorporados à API da JAI, podendo ser chamados em funções, também, como métodos nativos.

A ImageIO é uma biblioteca independente de plataforma que permite abrir e salvar imagens armazenadas no disco rígido. Ela trabalha com o uso de *plug-ins* de leitura e escrita que permitem manipular diversos formatos de imagens. Esses *plug-ins*, como o nome indica, podem ser adicionados à biblioteca, o que possibilita a sua extensão para suportar formatos de imagens mais novos.

Através da biblioteca ImageIO, são importadas as imagens que serão corrigidas. As imagens são lidas para dentro do programa e assim podem ser manipuladas as operações necessárias. Também é através dessa biblioteca que as imagens corrigidas ou os padrões de iluminação podem ser salvos.

3.2. Arquitetura

Os métodos de correção de iluminação de que trata este trabalho serão incorporados ao sistema de processamento e análise de imagens desenvolvido pelo Grupo de Processamento de Informação Multimídia – GPIM/UFSM. Este sistema possui agregadas várias ferramentas para pré-processamento, segmentação, e extração de características.

O desenvolvimento do Arthemis, segue o padrão arquitetural de camadas, descrito por Buschmann [6], no qual o usuário faz uma requisição à camada superior e essa requisição é tratada pelas camadas inferiores [6]. O resultado obtido em cada camada é repassado de volta, camada a camada, até ser apresentado ao usuário. Assim, o nosso sistema apresenta um camada central ou núcleo, uma camada de aplicação e uma de apresentação. O principal

objetivo dessa arquitetura é tornar o software flexível em relação às suas funcionalidades, decompondo-o em blocos de funções similares ao seu nível de abstração [6].

Consistentemente com a arquitetura do Artemis, o sistema de correção de iluminação foi estruturado em três camadas: a interface gráfica (primeira camada), os métodos (segunda camada) e núcleo (terceira camada). A interface gráfica possibilita a interação direta dos usuários com o sistema. Na camada intermediária encontram-se os métodos de geração do padrão de iluminação e de correção das imagens. Nela são recebidas as imagens a serem corrigidas e os parâmetros de operação. A partir dessa camada é acessada a camada núcleo onde, finalmente, as operações sobre os pixels são efetivadas utilizando a biblioteca JAI e os métodos criados e estendidos como nativos.

4. Resultados

Os métodos descritos nas seções anteriores foram implementados e utilizados com sucesso pelos membros do GPIM. Durante os testes, observou-se que tanto o desfoque gaussiano como a interpolação bicúbica apresentam resultados satisfatórios no que se refere à extração do padrão de iluminação nas situações em que cada um deles se aplica. Na Figura 4 é mostrado o resultado da correção de iluminação, onde o padrão de iluminação extraído por desfoque gaussiano é subtraído da imagem original.



Figura 4: Correção de vignetting por subtração do padrão de iluminação gerado por desfoque gaussiano.

Em comparação com outros sistemas de processamento de imagens, a ferramenta desenvolvida proporcionou um aumento de produtividade pela redução do tempo gasto pelos usuários para selecionar o tamanho do filtro gaussiano ou fazer a escolha correta dos pontos representativos do fundo.

5. Conclusão

Neste trabalho, dois métodos foram implementados para a extração do padrão de iluminação irregular em imagens digitais para a sua posterior aplicação na correção das imagens afetadas, a saber, o desfoque gaussiano e a interpolação bicúbica. Ambos os métodos necessitam de intervenção do usuário, que fornece

parâmetros ou seleciona pontos representativos do padrão de iluminação. Para isso foi desenvolvida uma interface gráfica que torna mais fácil e intuitiva a definição destes parâmetros, reduzindo o tempo gasto pelo usuário para fazer a escolha mais adequada dos dados entrada dos métodos.

A ferramenta desenvolvida será integrada a um sistema de processamento e análise de imagens, cuja arquitetura facilita a incorporação de novos componentes, melhorando a performance de métodos que se baseiam na medida de intensidades.

Como continuação deste trabalho, novos métodos de correção das imagens utilizando o padrão de iluminação extraído serão implementados e testados. Também a realização da correção em batch de conjuntos de imagens adquiridos sob as mesmas condições será adicionada ao sistema.

6. Referências

- [1] S. Inoué, K.R. Spring, *Video microscopy - The fundamentals*, 2nd Ed., New York, Plenum Press, 1997.
- [2] J. C. Russ, *Image Processing Handbook*, CRC, Boca Raton, 1998.
- [3] F J W-M Leong, M Brady and J O'D McGee, "Correction of uneven illumination (vignetting) in digital microscopy images", *J. Clin. Pathol.* 56; 2003, pp. 619-621.
- [4] M. Wang, J. Pan, S. Chen, H. Li, "A Method of Removing the Uneven Illumination Phenomenon for Optical Remote Sensing Image", *Proceedings of the Geoscience and Remote Sensing Symposium*, 2005, IEEE International, Volume 5, Issue , 25-29, 2005 pp. 3243-3246.
- [5] W. H. Press, S. A. Teukolsky, W. T Vetterling, *Numerical recipes in C++: the art of scientific computing*, 3rd. Ed. , Cambridge University Press, 2007.
- [6] Sun Microsystems, *Programming in Java Advanced Imaging. (JAI Guide)*.
- [7] F. Buschmann, R. Meunier, *Pattern - Oriented Software Architecture: A System of Patterns*, Vol. 1, John Wiley & Sons, New York, 1996.