JAVA ADVANCED IMAGING, UMA APLICAÇÃO EM PROCESSAMENTO DIGITAL DE IMAGENS

Fagner Fernandes Santos[1], Leandro Luiz de Almeida[1]

[1]Faculdade de Informática – Universidade do Oeste Paulista (UNOESTE), Presidente Prudente, SP – Brasil. fagner.ffs@gmail.com, llalmeida@unoeste.br

RESUMO

Esse trabalho traz um estudo de técnicas de Processamento Digital de Imagens e Fotogrametria que, aplicadas em conjunto com os recursos oferecidos pela linguagem JAVA e sua API Java Advanced Imaging , permitem a geração de imagens de alta resolução a partir de uma seqüência de imagens adquiridas por um sensor de baixa resolução, a fim de minimizar a falta de qualidade na determinação dos alvos de interesse que necessitam ser analisados. A API Java Advanced Imaging (JAI API) amplia o alcance da plataforma Java para sofisticar e aumentar a performance da função de processamento de imagens incorporadas às Aplicações Java.
Palavras-chaves: Processamento de imagens; fotogrametria; java

JAVA ADVANCED IMAGING, AN APPLICATION IN DIGITAL IMAGE PROCESSING

ABSTRACT

This work presents a study of techniques of Digital Image Processing and Photogrammetry which, applied together with the features offered by JAVA and API Java Advanced Imaging, allow the generation of high-resolution images from a sequence of images acquired by a low-resolution sensor in order to minimize the lack of quality in determining the targets of interest that need to be analyzed. The Java Advanced Imaging API (JAI API) extends the reach of the Java platform to refine and enhance the performance of image processing function incorporated into Java applications.
Keywords: Image processing, fotogrammetry; java.

1. INTRODUÇÃO

Imagens com um baixo nível de resolução podem dificultar a visualização dos fenômenos imageados. Isso pode ser um problema em várias áreas do conhecimento que utilizam análise de imagens, as quais, com imagens de baixa resolução, podem levar a uma decisão sem a precisão esperada.

Segundo Almeida (2001) alguns dos fatores que podem causar um baixo nível de resolução em uma imagem são:

A resolução do sensor;

- Equipamento que irá adquirir a imagem não apresenta uma boa qualidade ou recursos especiais que facilitam a obtenção da mesma;
- Distorção ou aberração das lentes da câmara;
- Durante a obtenção da imagem pode ocorrer um certo deslocamento do equipamento, que afeta a qualidade da imagem;
- Efeitos naturais, como excesso ou carência de luz.

Devido a esses fatores, busca-se através de aplicação de técnicas de Processamento Digitais de Imagens e Fotogrametria, conciliadas com as funcionalidades oferecidas pela API (Application Programming Interface) Java Advanced Imaging, minimizar a falta de qualidade na determinação dos alvos de interesse que necessitam ser analisados.

Esse trabalho traz um estudo sobre a análise de seqüências de imagens de uma mesma cena, com o objetivo de se criar uma imagem com uma melhor resolução, a partir da transformação e fusão de várias imagens, utilizando-se para isto os recursos oferecidos pela linguagem Java e sua API Java Advanced Imaging.

Java Advanced Imaging é uma API lançada recentemente pela SUN Microsystems, que estende a plataforma Java (incluindo a API Java 2D) incorporando nas Aplicações recursos sofisticados e de alta performance para Processamento de Imagens.

O método utilizado neste trabalho, na verdade, trata de uma técnica de Restauração, uma vez que se tenta estimar a função imagem original, com base na taxa de sobreposição existente em uma seqüência de imagens pertencentes a uma mesma cena.

Existem vários exemplos de seqüências de imagens de baixa resolução, nas quais necessita-se observar detalhes que não são possíveis sem um tratamento adequado aplicado à mesma, por exemplo, uma seqüência de imagens geradas por câmeras de circuito fechado onde há a necessidade de identificar a face de pessoas suspeitas.

Por outro lado, quando se dispõe de uma seqüência de imagens, em posições próximas, seria possível realizar uma super amostragem, ou seja, aumentar a resolução geométrica dessa imagem para que os detalhes sejam definidos claramente (Almeida, 2001).

2. MÉTODOS PARA RESTAURAÇÃO A PARTIR DE SEQÜÊNCIAS

Peleg et al. (1987) destacam que, a partir de imagens de baixa resolução é possível produzir uma imagem com resolução melhorada, usando os deslocamentos conhecidos entre as imagens. O processo para a geração da imagem de alta resolução, proposto pelos autores, inicia-se com uma imagem superamostrada, determinada como sendo uma aproximação inicial.

Este processo é iterativo, onde os novos valores para cada pixel são determinados, nesta imagem, após se verificar o menor erro ocasionado na diferença entre os níveis de brilho

dos pixels das imagens de baixa resolução e os seus correspondentes na imagem de alta que está sendo estabelecida. Varia-se a intensidade de brilho dos pixels que estão sendo analisados na imagem superamostradas, em (-1, 0, +1), realizando-se combinações neste bloco, com o intuito de estabelecer o brilho gerado pela menor diferença entre as imagens.

Peleg e Irani (1991) sugeriram melhorias para o método apresentado por Peleg et al. (1987). Os autores tomaram o registro de imagens como base para uma boa determinação de uma imagem de melhor resolução. Desenvolveram algoritmos iterativos para se obter um melhoramento na resolução das imagens com precisão subpixel, tanto em imagens de baixa resolução em níveis de cinza como nas coloridas. Com os deslocamentos da imagem e o processo de imageamento conhecidos, pode-se realizar uma fusão entre as imagens de baixa resolução, desde que pertençam à mesma seqüência que compõe a cena, usando técnicas de interpolação e gerando-se, assim, uma imagem de melhor resolução.

Baker & Kanade (1999) apresentam uma solução para a geração de imagens de super resolução, a partir de seqüências de imagens não planares, não rígidas e sujeitas auto-oclusões, já que, a grande maioria dos trabalhos desenvolvidos, nessa linha, utiliza imagens planares e rígidas. Os autores buscaram melhorar a resolução das imagens, ou partes das mesmas, que envolviam expressões faciais, pois há grande dificuldade na construção do modelo 3-D da face humana. Com base no algoritmo do fluxo ótico (algoritmo para a determinação dos parâmetros de registro e retificação entre imagens), versão de Lucas & Kanade (1981) apud Baker & Kanade (1999), consegue-se gerar, simultaneamente, os parâmetros para retificação e registro das imagens, e a de super resolução.

2.1. Registro de Imagens

Segundo Lillesand & Kiefer (1987), no processo de aquisição de imagem podem ocorrer degradações ou distorções que podem ser causadas pela variação na iluminação no momento da aquisição da imagem, devido às condições ambientais, o local em que o sensor se encontra, além, das características do mesmo.

Para correção dessas degradações ou distorções, devem ser feitas a restauração e retificação das imagens, mas, para se aplicar qualquer tipo de técnica para a restauração dessas imagens é necessário, anteriormente, realizar a retificação das imagens que pertencem à cena adquirida, para se posicionar uma imagem em relação à outra, ou seja, o registro de imagens deve ser executado.

O registro de imagens compreende uma transformação geométrica que relaciona cada ponto de uma imagem com as coordenadas de um sistema de referência qualquer. Esse sistema de referência é, em última instância, o sistema de coordenadas planas de uma certa projeção cartográfica, ou ainda, uma outra imagem que fornecerá os pontos de controle para que as outras que fazem parte da cena possam ser posicionadas em relação a esta que foi determinada como a de referência (Almeida, 2001).

O esquema demonstrado na figura 1 demonstra as etapas para se executar o registro de imagens:

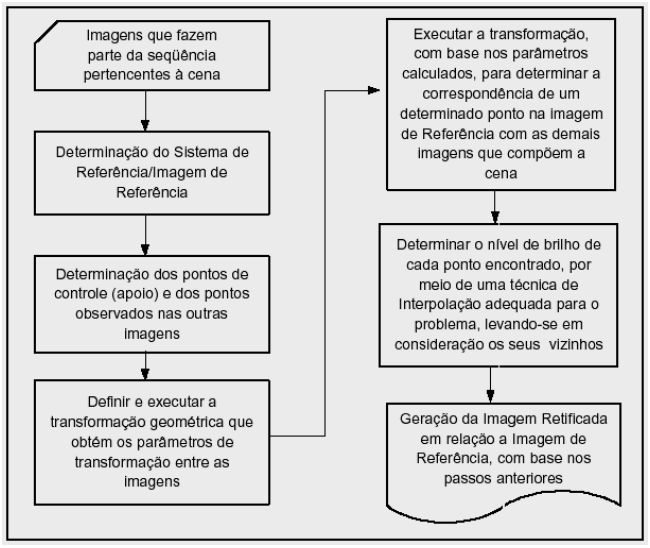

Figura 1 – Esquema geral para execução do Registro de Imagens (Fonte: Almeida, 2001).

O processo de *matching* entre imagens é a principal etapa do registro de imagens. Consiste em determinar a correspondência entre os pontos da imagem de referência com os de uma outra imagem que pertence à cena para que seja possível se retificar todas as imagens pertencentes à mesma cena em relação a um referencial qualquer, que neste caso, será uma das imagens.

No caso de uma região relativamente plana, topograficamente, é possível determinar o registro aplicando-se uma transformação de perspectiva, e até mesmo uma Transformação Afim, mas devido à paralaxe existente em uma imagem, por apresentar alguma forma de relevo (topografia), não se consegue determinar o registro com apenas uma transformação para toda a cena, pois, dependendo da distância do sensor ao alvo, este pode apresentar formas diferentes (Almeida, 2001).

Segundo Richards (1986) duas imagens podem ser registradas uma a outra através do registro de cada uma delas, separadamente, em relação às coordenadas de controle definidas previamente. Uma possibilidade seria a de escolher uma imagem como referência (imagem principal) e registrar às outras imagens (imagens secundárias) em relação à de referência. Nestes casos, na utilização das técnicas de correção geométrica, as coordenadas (x, y) correspondem às coordenadas da imagem que será registrada (secundária).

2.2. Transformações Geométricas

Transformações são operações que transformam coordenadas de um determinado sistema de referência para outro qualquer, com base nos pontos de controle determinados previamente (Almeida, 2001).

Na realização do registro tradicional, essas transformações são determinadas usando os pontos de controle adquiridos do processo de correspondência, que neste trabalho, foi realizado manualmente.

Matematicamente, pode-se dizer que dados *n* pontos de controle em duas imagens de uma mesma cena, *[(Xi, Yi), (xi, yi)]*, onde *i=1,...,n*, a transformação realizará o mapeamento dos pontos entre as imagens, podendo ser definida genericamente, como em (2.1):

$$\begin{aligned} Xi &= f(xi, yi) \\ Yi &= g(xi, yi) \end{aligned} \qquad (2.1)$$

onde *(xi, yi)* corresponde às coordenadas do ponto imagem na primeira imagem, e *(Xi, Yi)* corresponde às coordenadas do mesmo ponto na segunda imagem, sendo esta, a imagem que possui os pontos de controle.

Segundo Brito & Coelho (2002), a transformação afim pode ser utilizada para a retificação aproximada de uma imagem. Consiste em conhecendo-se as coordenadas de no mínimo três pontos não-colineares no sistema de coordenadas da imagem inicial e no sistema de coordenadas da imagem final, através de um ajustamento pelo método paramétrico, calcular os coeficientes de transformação entre ambos os sistemas.

Outros tipos de transformação também são utilizados para este fim, tais como, a

transformação projetiva e a transformação perspectiva.

A condição projetiva entre dois planos é determinada pelas equações lineares fracionárias que são compostas por oito parâmetros de transformação, conforme (2.3) (adaptadas de Moffitt & Mikhail, 1980).

$$X_i = \frac{a_1 * x_i + b_1 * y_i + c_1}{a_0 * x_i + b_0 * y_i + 1}$$

(2.3)

$$Y_i = \frac{a_2 * x_i + b_2 * y_i + c_2}{a_0 * x_i + b_0 * y_i + 1}$$

Nas equações anteriores:

Xi, Yi: coordenadas no plano objeto ou na imagem de referência;

xi, yi: coordenadas no plano imagem;

a0, a1,..., b0, b1,...: parâmetros de transformação.

2.3. Métodos de Interpolação

O grande problema da aplicação de transformações em uma imagem encontra-se na determinação exata do tom de cinza a ser destinado aos *pixels* da nova imagem, como mostra a figura 2. Algumas transformações podem causar efeitos indesejáveis na imagem, ou seja, apresentar baixa resolução radiométrica (Brito e Coelho, 2002).

Por isso faz-se necessária a interpolação, o qual, dependendo da técnica utilizada, poderá causar um efeito de suavização na imagem e para que os novos *pixels* tenham a cor que deveriam ter por estarem em tal posição.

A outra possibilidade de se utilizar uma técnica de interpolação, é para a criação de uma imagem superamostrada, ou seja, uma imagem ampliada em relação a sua original.

As técnicas de interpolação mais utilizadas são: Vizinho Mais Próximo, Interpolação Bilinear e Interpolação Bicúbica (Almeida, 2001).

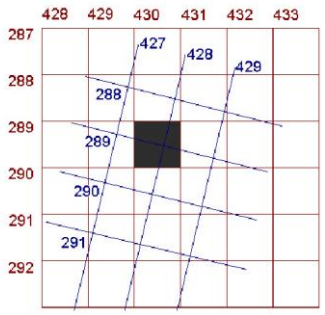

Figura 2 – Imagem original encontra-se representada com seu grid de pixels vermelho. Nova imagem encontra-se representada através do quadriculado azul sobreposto.

Na técnica de interpolação pelo vizinho mais próximo o nível de cinza de um ponto transformado é igual ao nível de cinza do *pixel* mais próximo da imagem digital. No caso da interpolação bilinear, o nível de cinza de um ponto transformado é obtido por interpolação dos quatro *pixels* mais próximos. Já a interpolação bicúbica utiliza os dezesseis *pixels* mais próximos de um ponto transformado para interpolação do nível de cinza.

3. GERAÇÃO DE IMAGENS DE MELHOR RESOLUÇÃO

A geração de imagens de melhor resolução com base em uma seqüência de imagens necessita, primeiramente, que as imagens pertencentes à cena estejam retificadas, ou seja, o registro de imagens deve ter sido executado, e por isso, é considerado como a etapa primeira neste processo.

Uma visão geral da metodologia empregada para o desenvolvimento do trabalho pode ser vista na figura 3.

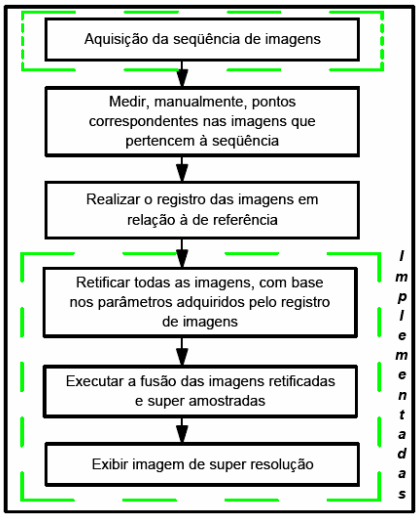

Figura 3 – Visão geral das etapas para a geração das imagens de melhor resolução

3.1. Etapas para a geração das Imagens

3.1.1. Coleta das Imagens

Para fim de experimento, as imagens podem ser coletadas introduzindo-se pequenos deslocamentos lineares de forma a obter seqüências de imagens. Esta coleta pode ser realizada também, de forma contínua por meio de câmeras de vídeo, mas neste caso será necessária a utilização de software específico para digitalização e separação dos frames de interesse;

3.1.2. Registro e Retificação

Para se obter o registro e, posteriormente, a retificação das imagens, é necessário que seja feita uma medição manual dos pontos correspondentes entre as imagens. Para isso, pode ser utilizado um software como Adobe Photoshop, onde deverão ser medidos vários pontos em todas as imagens, determinando suas correspondências em relação às demais, utilizando o máximo nível de zoom do software e anotando as coordenadas linha e coluna, com o objetivo de se minimizar o erro de pontaria.

Essas coordenadas (em *pixels*) devem ser transferidas para o referencial fotogramétrico (em milímetros) pelas equações descritas em 3.1. Para os cálculos pode ser utilizado o software Microsoft Excel.

$$x_{fot} = (x_i - NCol/2) * TamPixel \quad (3.1)$$

$$y_{fot} = -(y_i - NLin/2) * TamPixel$$

Onde:

xfot, yfot: coordenada no referencial fotogramétrico;

xi, yi: coordenada imagem (coluna, linha);

NCol: número de colunas da imagem adquirida;

NLin: número de linhas da imagem adquirida;

TamPixel: tamanho do *pixel* em milímetros na imagem a ser gerada.

Com estes pontos determinados, o próximo passo é determinar os parâmetros para a realização do registro entre as imagens. Para isso, é utilizada a equação da transformação Projetiva.

Após os parâmetros terem sido determinados, faz-se simultaneamente, a retificação e reamostragem das imagens que pertencem à seqüência em relação à imagem de referência. Esta retificação é realizada obtendo-se a correspondência de cada *pixel* da imagem de referência superamostrada com a imagem original de baixa resolução, que inicialmente, possuí intensidade de brilho igual a zero.

Com base na equação da transformação projetiva que possibilita o registro das imagens, determina-se à correspondência dos *pixels* nas imagens de baixa resolução.

Como as coordenadas projetadas na imagem de baixa resolução geralmente estão em frações do *pixel*, há a necessidade de uma interpolação entre os vizinhos para a determinação do nível de brilho relativo ao que se deseja determinar na imagem de super resolução.

O método de interpolação bilinear é o utilizado, levando-se em consideração a influência de cada *pixel* em relação ao que deveria ser gerado.

3.1.3. Fusão e geração da Imagem de Super Resolução

A etapa final do processo é a fusão dos *pixels* correspondentes nas imagens retificadas que pertençam à cena em relação à de referência. Nesta etapa obtém-se a imagem de super resolução, podendo ser realizada de duas formas:
i.) Média entre os valores de brilho dos *pixels*: é realizada a partir dos *pixels* correspondentes determinados pelos processos de registro e retificação para as imagens da seqüência, para se determinar o nível de brilho do *pixel* na imagem resultado, conforme mostra a figura 4.

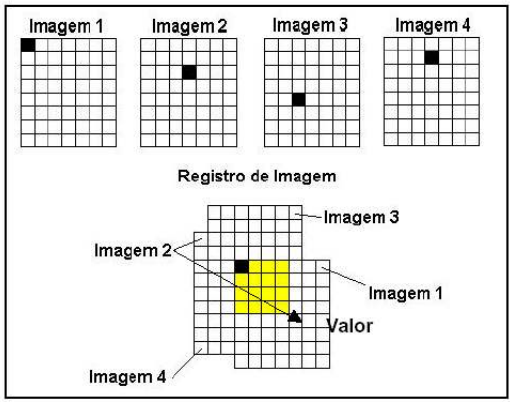

Figura 4 – Fusão pelo método da Média entre os pixels das imagens que pertencem à cena (fonte: Almeida, 2001).

ii.) Processo com minimização de erros: inicialmente realiza-se uma primeira estimação da imagem de alta resolução, por exemplo, a imagem de referência superamostrada pela interpolação bilinear. A partir da imagem de alta resolução aproximada, projeta-se cada *pixel* nas imagens de baixa resolução.

O valor projetado será fracionário (ao nível de *subpixel*) e deve-se interpolar o valor correspondente em cada imagem de baixa resolução. Com isso, obtém-se a diferença entre o valor projetado e o valor observado, originando um erro.

Calcula-se o erro médio quadrático (equação 3.2) para o bloco de quatro *pixels*, considerando-se todas as imagens da seqüência; varia-se para -1 o valor de brilho do primeiro *pixel* na imagem de alta resolução e repete-se a projeção para o cálculo do erro; varia-se para +1 e repete-se o cálculo. Por meio desse processo, calculam-se os erros para todas as combinações possíveis de valores de brilho, variando no intervalo (-1, 1). A combinação que apresentar o menor erro médio quadrático é mantida na imagem de alta resolução.

O Erro Médio Quadrático é dado pela equação:

$$e = \sqrt{\frac{\sum r^2}{n-1}} \qquad (3.2)$$

Onde:

e: Erro Médio Quadrático;

r: resíduo (diferença, em intensidade de brilho, entre o *pixel* projetado e o observado);

n: quantidade de *pixels* da Imagem.

4. JAVA ADVANCED IMAGING

Um requisito básico para se desenvolver um sistema para manipulação de imagens, são as primitivas gráficas que a linguagem de programação oferece.

Java tem evoluído ao longo do tempo desde sua concepção. As primeiras primitivas gráficas vieram no Abstract Windowing Toolkit (AWT), como o desenho de formas geométricas básicas (retângulo, círculos, ...), tratamento de cor, etc. Mais recentemente, a SUN liberou um novo pacote gráfico, o Java2D API (Application Programming Interface), oferecendo novos recursos com melhorias em gráficos 2D, textos e algumas funções para processamento de imagens matriciais.

Para a área de processamento de imagens digitais, existe um projeto mais avançado, o Java Advanced Imaging (JAI API), recomendado para áreas como:

- Defesa militar e serviços de inteligência;
- Processamento digital de imagens de satélite;

- Bioinformática;
- Pesquisa aplicada;
- Fotografia digital;
- Comércio eletrônico e ensino.

Dentre as funções de processamento de imagens encontram-se funções de convolução, cubo de cores, transformações geométricas, histograma, espaço de cores IHS (Intensity Hue Saturation) entre outros, interpolações bicúbica, bilinear, operação como interpolação pelo vizinho mais próximo, operação pontual para transformação de valores de *pixels*, algoritmos de morfologia matemática, entre outras opções.

Na ferramenta desenvolvida houve a necessidade de abrir a seqüência de imagens adquiridas da cena para aplicar a metodologia estudada. Para isso há a necessidade de se trabalhar com vários formatos de imagens. A API Java Advanced Imaging forneceu recursos para facilitar e melhorar este processo, pois com a AWT e a API Java2D seria mais custoso e menos eficiente realizar esta tarefa.

Foi utilizada na ferramenta, a estrutura multi-bandas para a manipulação das imagens, ou seja, cada componente do padrão de cores RGB em uma banda, sendo a imagem de 24 bits (adotado esse padrão no projeto desenvolvido), um byte por banda, como mostra a figura 5.

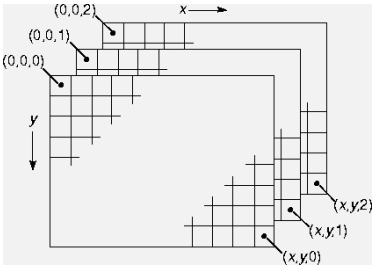

Figura 5 – Estrutura de Imagens multi-bandas

A classe *RendereOp* é uma classe da JAI (Java Advanced Imaging API) e fornece esse recurso e outros para a manipulação de imagens. Também é possível com esse tipo de Imagens aplicar os métodos existentes na JAI para processamento de imagens.

O código abaixo mostra um exemplo de conversão de uma Imagem para Escala de Cinza, utilizando uma matriz de combinação, recurso existente na JAI:

```
//imagem é do tipo RenderedOp e representa a
//imagem que será convertida para escala de
cinza
SampleModel sm = imagem.getSampleModel();
double matComb = {
  { .299D, 0.587D, 0.114D, 0.0D },
  { .299D, 0.587D, 0.114D, 0.0D },
  { .299D, 0.587D, 0.114D, 0.0D },
};
ParameterBlock pb = new ParameterBlock();
pb.addSource(this.imagem);
pb.add(matComb);
RenderedOp result = JAI.create("bandcombine",
pb, null);
//result é a imagem resultante em escala de cinza
```

Com isso é possível efetuar várias outras operações que envolvam uma matriz de combinação, alterando somente os valores da matriz.

Para transformações geométricas, como Translação, Escala e Rotação, o código abaixo mostra um exemplo de um dos recursos oferecidos pela JAI:

```
//imagem é do tipo RenderedOp e representa a
// imagem que será aplicada a transformação
ParameterBlock pb = new ParameterBlock();
pb.addSource(imagem);
pb.add(ex);              // Escala para x
pb.add(ey);              // Escala para y
pb.add(tx);              // Translação para x
pb.add(ty);              // Translação para y
//Interpolação Bilinear
pb.add(new InterpolationBilinear());
// Efetua a operação de escala
RenderedOp resultado;
```

```
resultado = JAI.create("scale", pb, null);
```

Neste exemplo, a transformação é feita de acordo com os parâmetros definidos para ex, ey (escala), tx, ty (translação). No exemplo também é aplicada uma interpolação bilinear em toda a imagem resultante do processo, sendo facilmente configurada a interpolação e aplicada pelo método da JAI junto com a transformação.

O tipo de estrutura de imagens da JAI *RenderedOp* não pode ser utilizado diretamente com os métodos da AWT, pois o tipo principal de estrutura de imagens que a AWT trabalha é o *Image*, mas isso pode ser obtido através de conversões de tipo e métodos oferecidos pela JAI, como o exemplo mostrado a seguir:

```
//img é do tipo RenderedOp e representa a
imagem
//que será convertida para o formato Image da
//AWT, visto nas duas linhas seguintes
BufferedImage bimg = img.getAsBufferedImage();
Image imagemconv = (Image) bimg;
//Abaixo o processo inverso
ParameterBlock pb = new ParameterBlock();
 pb.add(imagemconv);
 img = JAI.create("awtImage", pb);
```

Com isso, no desenvolvimento da ferramenta foi possível implementar todos os passos utilizando a linguagem Java e os recursos da JAI, melhorando assim o processamento de imagens sem deixar de utilizar os componentes Java Swing e da AWT.

5. CONCLUSÕES

As melhorias alcançadas pela ferramenta desenvolvida no projeto são baseadas não somente em filtros em cada imagem, mas sim pela análise feita de uma mesma região em mais de uma imagem, sendo que um determinado *pixel* que não se apresenta com qualidade em uma imagem seja compensado por outro que se localiza em outra imagem da mesma cena.

Com isso, a imagem gerada como resultado final pode apresentar características que não poderiam ser observadas, ou que foram melhoradas, e também imagens superamostradas, em relação às imagens originais de baixa resolução, ressaltando que as informações das imagens, como o nível de brilho dos *pixels* compensados, são baseadas em dados reais contidas nas imagens de baixa resolução da mesma cena.

A utilização da API Java Advanced Imaging trouxe vantagens para a implementação da ferramenta e também foram obtidos resultados satisfatórios quanto ao desempenho e eficiência. Essa API fornece todo o suporte necessário para que se possa trabalhar com Processamento de Imagens com maior eficiência.

REFERÊNCIAS

ALMEIDA, L. L., Geração de Imagens de melhor Resolução à partir de Seqüências de Imagens. Presidente Prudente, 2001. 132 p. Dissertação de Mestrado – Faculdade de Ciências e Tecnologia, Campus de Presidente Prudente, Universidade Estadual Paulista.

BAKER S.; KANADE T. Super-Resolution Optical Flow. [s.1.]: CMU-RI-TR-99-36, 1999.

BRITO, J.; COELHO, L., Fotogrametria Digital. Instituto Militar de Engenharia, Rio de Janeiro, 2002.

FONSECA, L. M. G.; MANJUNATH, B. S. Registration Techniques for Multisensor Remotely Sensed Imagery; Photogrammetric Engineering & Remote Sensing, Vol. 62, nº 9, September 1996, pp. 1049-1056.

LILLESAND, T. M.; KIEFER, R. W. Remote Sensing and Image Interpretation; New York. John Wiley & Sons. 2ed. 1987.

MOFFITT, F. H.; MIKHAIL, E. M. Photogrammetry. 3ª ed. Vol. II. New York, Harper & Row Publishers, 1980. 268p.

PELEG, S., IRANI, M. “Improving Resolution by Image Registration.” Graphical Models and Image Processing. Vol. 53, nº 3, 1991, p. 231-239.

PELEG, S.; KEREN, D.; SCHWEITZER, L. Improving Image Resolution Using Subpixel Offsets (Motion). Pattern Recognition Letters, 1987, 223-226.

RICHARDS, J. A. Remote Sensing Digital Image Analysis – In Intruduction. Berlin-Heidelberg. Springer-Verlag, 1986.

DEITEL, H. M.; DEITEL, P. J. Java Como Programar. 4.ed. Porto Alegre: Bookman, 2003.

MARQUES, O. F.; VIEIRA, H. N. Processamento Digital de Imagens. Rio de Janeiro: Brasport, 1999.

SUN. Programming in Java Advanced Imaging, Release 1.0.1, 1999. Disponível em: <http://java.sun.com/products/java-media/jai/forDevelopers/jai1_0_1guide-unc/JAITOC.fm.html.> Último acesso em: 16 ago. 2004.