UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS AGRONÔMICAS
CAMPUS DE BOTUCATU

DESENVOLVIMENTO DE UM APLICATIVO PARA PROCESSAMENTO DE IMAGENS OBTIDAS POR FOTOGRAFIAS AÉREAS

WILIAM CARLOS GALVÃO

Tese apresentada à Faculdade de Ciências Agronômicas da UNESP – Campus de Botucatu, para obtenção do título de Doutor em Agronomia (Energia na Agricultura)

BOTUCATU - SP
Dezembro – 2015

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS AGRONÔMICAS
CAMPUS DE BOTUCATU

DESENVOLVIMENTO DE UM APLICATIVO PARA PROCESSAMENTO DE IMAGENS OBTIDAS POR FOTOGRAFIAS AÉREAS

WILIAM CARLOS GALVÃO

ORIENTADOR: PROF. DR. ZACARIAS XAVIER DE BARROS

Tese apresentada à Faculdade de Ciências Agronômicas da UNESP – Campus de Botucatu, para obtenção do título de Doutor em Agronomia (Energia na Agricultura)

BOTUCATU - SP

Dezembro – 2015

FICHA CATALOGRÁFICA ELABORADA PELA SEÇÃO TÉCNICA DE AQUISIÇÃO E TRATAMENTO DA INFORMAÇÃO - DIRETORIA TÉCNICA DE BIBLIOTECA E DOCUMENTAÇÃO - UNESP - FCA - LAGEADO - BOTUCATU (SP)

Galvão, Wiliam Carlos, 1977-
G182d Desenvolvimento de um aplicativo para processamento de imagens obtidas por fotografias aéreas / Wiliam Carlos Galvão. - Botucatu : [s.n.], 2015
ix, 69 f. : fots. color.; grafs. color., ils. color., tabs.

Tese (Doutorado) - Universidade Estadual Paulista, Faculdade de Ciências Agronômicas, Botucatu, 2015
Orientador: Zacarias Xavier de Barros
Inclui bibliografia

1. Imagem bidimensional. 2. Imagem tridimensioanl. 3. Processamento de Imagens. I. Barros, Zacarias Xavier de. II. Universidade Estadual Paulista "Júlio de Mesquita Filho" (Câmpus de Botucatu). Faculdade de Ciências Agronômicas. III. Título.

unesp

UNIVERSIDADE ESTADUAL PAULISTA

CAMPUS DE BOTUCATU

FACULDADE DE CIENCIAS AGRONOMICAS DE BOTUCATU

CERTIFICADO DE APROVAÇÃO

TÍTULO: DESENVOLVIMENTO DE UM APLICATIVO PARA PROCESSAMENTO DE IMAGENS OBTIDAS POR FOTOGRAFIAS AÉREAS

AUTOR: WILIAM CARLOS GALVAO

ORIENTADOR: Prof. Dr. ZACARIAS XAVIER DE BARROS

Aprovado como parte das exigências para obtenção do Título de DOUTOR EM AGRONOMIA (ENERGIA NA AGRICULTURA) , pela Comissão Examinadora:

Prof. Dr. ZACARIAS XAVIER DE BARROS
Dep de Engenharia Rural / Faculdade de Ciencias Agronomicas de Botucatu

Profa. Dra. BRUNA SOARES XAVIER DE BARROS
Depto de Engenharia Rural - FCA

Prof. Dr. PAULO ROBERTO ARBEX SILVA
Dep de Engenharia Rural / Faculdade de Ciencias Agronomicas de Botucatu

Prof. Dr. ADELSON FRANCISCO MAIA JUNIOR
Centro de Ciencias Exatas - UNIESP/BAURU

Prof. Dr. ELVIO GILBERTO DA SILVA
Centro de Ciencias Exatas - USC/BAURU

Data da realização: 09 de dezembro de 2015.

"Dêem-me uma alavanca e um ponto de apoio e eu moverei o mundo".
(Arquimedes)

“Combati o bom combate, acabei a corrida, guardei a Fé”. (II Timóteo 4:7)

AGRADECIMENTOS

Agradeço a Deus, primeiramente, por sempre guiar meus passos, por me dar coragem para lutar, forças para enfrentar os desafios e perseverança para vencer;

À minha família em especial a meus pais, Maria e Sérgio, a minha esposa Flávia, a minha filha Isabelle, que me apoiaram incondicionalmente e incansavelmente acreditaram em meu potencial;

Ao Professor Doutor Ângelo Catâneo (*in memórian*) pela oportunidade dada;

À todas as pessoas que me auxiliaram na realização desta tese, em especial, ao meu orientador Professor Doutor Zacarias Xavier de Barros, que me acolheu e em todas as etapas me guiou e orientou de forma muito incentivadora e pacienciosa;

Aos meus colegas de pós-graduação Flávia "minha esposa", Elvio Gilberto da Silva, Bruna Soares Xavier de Barros, Ronaldo Pollo pelas valiosas sugestões e amizade;

À Coordenação do Programa de Pós-Graduação em Agronomia: Energia na Agricultura, da Faculdade de Ciências Agronômicas da UNESP, campus de Botucatu - SP.

SUMÁRIO

Página

LISTA DE FIGURAS

Página

LISTA DE TABELAS

1 RESUMO

O processamento de imagens é uma área em crescimento. Diversos temas científicos são abordados e em alguns casos de caráter interdisciplinar. Entre eles pode-se citar: obtenção de imagens, compreensão de imagens, análise em multi-resolução e em multi-frequência, análise estatística, codificação e a transmissão de imagens.

O conhecimento e a utilização dos recursos naturais, humanos e econômicos de uma região ou de um país, é de fundamental importância para o seu desenvolvimento. O planejamento do uso do solo tem se tornado cada vez mais importante diante desse contexto. Na agronomia mais precisamente no estudo de fotografias aéreas pode-se relacionar vários setores que poderão ser beneficiados por esse estudo, como por exemplo: levantamento e delimitação de bacias hidrográficas, geoprocessamento, fotogrametria, fotopedologia, fotointerpretação, mapeamento de solos, geoestatística e ocupação e uso do solo.

Esse trabalho envolveu a linguagem de programação orientada a objetos JAVA e banco de dados relacional MYSQL ambos com licença gratuita e codigo livre, tornando possível o desenvolvimento de um sistema consistente, ganhando velocidade e confiabilidade na escrita dos códigos.

Este trabalho teve como objetivo o desenvolvimento de um programa de computador para: processamento de imagens de fotografias aéreas, obtendo-se por meio de imagens bidimensionias imagens estéreoscopicas tridimensionais.

DEVELOPMENT OF AN APPLICATION FOR IMAGE PROCESSING OBTAINED BY AERIAL PHOTOGRAPHS.Botucatu, 2015, 76 p.
Tese (Doutorado em Agronomia/Energia na Agricultura) - Faculdade de Ciências Agronômicas, Universidade Estadual Paulista.
Author: WILIAM CARLOS GALVAO
Advisor: ZACARIAS XAVIER DE BARROS

2 SUMMARY

The image processing is a growing area. Several scientific issues are addressed and in some cases interdisciplinary. These include: imaging, image understanding, analysis of multi-resolution and multi-frequency, statistical analysis, coding and transmission of images among others. The knowledge and use of natural resources, human and economic of a region or a country, it is of fundamental importance for its development. The planning of land use has become increasingly important in light of that context. In agronomy more precisely in the study of aerial photographs can relate various sectors that can benefit in this study, such as: survey and delineation of watersheds, GIS, photogrammetry, photo pedology, photo interpretation, soil mapping, geostatistics and occupation and use of ground. This work involved oriented programming language JAVA objects and relational database MySQL with both open source license, or free, making it possible to develop a consistent system, gaining speed and reliability in writing codes. This study aims to develop a computer program for: aerial photographs of image processing to get through two-dimensional images three-dimensional images.

Keywords: two-dimensional images, three-dimensional images. Image processing.

3 INTRODUÇÃO

A imagem pode conter mais informações do que um contexto descritivo, porém essas informações necessitam ser evidenciadas por meio de técnicas precisas, auxiliando no processo de investigação, análise e interpretação.

As primeiras classificações de uso do solo são baseados em trabalhos de campo. A partir de década de 50, pesquisadores em várias partes do mundo têm se dedicado à identificação e detalhamento de culturas agrícolas e outras informações por meio de fotografias aéreas.

A interpretação de fotografias aéreas é um recurso básico e constituído por técnicas de refinamento.

Na agronomia podemos relacionar vários setores que poderão ser beneficiados por esse estudo, como por exemplo: levantamento e delimitação de bacias hidrográficas, geoprocessamento, fotogrametria, fotopedologia, fotointerpretação, mapeamento de solos, geoestatística e ocupação e uso do solo.

O processamento de imagens origina-se do processamento de sinais. Os sinais, como as imagens, são um suporte físico que carrega no seu interior uma determinada informação. Esta informação pode estar associada a uma medida (neste caso falamos de um sinal em associação a um fenômeno físico), ou pode estar associada a um nível cognitivo (neste caso falamos de conhecimento). Processar uma imagem consiste em extrair mais facilmente a informação nela contida. Neste caso é possível fazer uma

comparação entre o processamento de imagem e área de computação gráfica, técnica que encontramos frequentemente aplicadas através de sequências animadas na televisão ou em filmes de cinema. A Computação Gráfica parte de uma informação precisa para obter uma imagem ou um filme.

Análise quantitativa e a interpretação de imagens representa atualmente um ponto de apoio importante em diversas disciplinas científicas podendo ser utilizada como apoio a interpretação. Cita-se por exemplo, na ciência dos materiais, na biofísica, na medicina, na física da matéria condensada.

A diversidade de aplicações do processamento de imagens está associada diretamente a análise da informação. Pois em todas estas disciplinas busca-se informações quantitativas que representem um fenômeno estudado. Quando observa-se do ponto de vista da ótica, uma imagem é um conjunto de pontos que converge para formar um todo, mas podemos dizer de uma maneira mais ampla que uma imagem é o suporte para efetuarmos troca de informações.

O termo imagem estava inicialmente associado ao domínio da luz visível, porém atualmente é muito frequente ouvirmos falar de imagens quando umas grandes quantidades de dados estão representadas sob a forma bidimensional (por exemplo: as imagens acústicas, sísmicas, de satélites, infravermelhas, magnéticas entre outros).

Este trabalho teve como objetivo o desenvolvimento de um programa de computador para: armazenamento, recuperação, processamento de imagens de fotografias aéreas, para obtenção por meio de imagens bidimensionais imagens estéreospocica tridimensionais.

4 REVISÃO DA LITERATURA

A busca por uma melhor compreensão é uma característica encontrada em todas as ciências. Porém pode ser mais evidente em áreas especificas como, por exemplo, análise de imagens médicas, onde o objetivo principal é através de um programa computacional, possa ser observado, detalhe que o ser humano por si só não seja capaz.

Portanto evidencia-se a necessidade de uma maior dedicação aos estudos sobre tratamento e manipulação de imagens e consequentemente o desenvolvimento de programas computacionais específicos para a agricultura, que além de manter de forma digitalizada pode também ser recuperado e manipulado com facilidade, podendo ser aplicado dos conceitos mais básicos aos mais complexos.

Os Sistemas de Informações Geográficas (SIG), juntamente com materiais cartográficos, têm contribuído para a obtenção de dados hidrológicos, devido à sua flexibilidade e disponibilidade, consistindo de sistemas computacionais que permitem integrar diversas informações espaciais da bacia hidrográfica. Através dos SIG, a informação é armazenada digitalmente e apresentada visual ou graficamente, permitindo a comparação e a correlação entre informações.

4.1. Fotografias aéreas

Segundo Moreira (2003), as fotografias aéreas possuem uma grande diversidade de funções, podendo ser usadas no planejamento de áreas urbanas, na cartografia, controle de queimadas e nas atividades agrícolas.

As fotografias aéreas podem ser obtidas através de vários tipos de plataformas, tais como aviões convencionais, helicópteros, balões e veículos aéreos não Tripulados (VANT).

Segundo Anderson (1982), a imagens aéreas obtidas, através de câmeras digitais, facilitam a utilização e geração de fotografias, permanecendo no formato digital, evitando a revelação e processamento posterior. A interpretação de fotografias e filmagens aéreas, para determinar metas e parâmetros (por exemplo: tonalidade, tamanho, forma, padrão, declividade, posição geográfica, sombra e outros) nas atividades agrícolas, serve para resgatar a ordem cronológica das mudanças ocorridas ao longo do tempo nas áreas pesquisadas.

De acordo com Moreira (2003), na agricultura, as fotografias aéreas são utilizadas no mapeamento de culturas, na avaliação de áreas cultivadas, na detecção de áreas afetadas, em cadastros rurais e no mapeamento de solo. E requerem um cuidado todo especial na hora de captação e interpretação das imagens, em função do nível de exigência necessário para a fotointerpretação, tais como altura, posicionamento da câmera, resolução, tonalidade e cor, tamanho, textura e sombra.

4.2. Fotointerpretação

A fotointerpretação é o processo de análise visual de imagens fotográficas, sendo o ato de examinar as imagens com o objetivo de identificar o seu significado. Este processo envolve três fases; fotoleitura, fotoanálise e fotointerpretação (MOREIRA, 2003).

Segundo Marques (1999) os elementos de um sistema de processamento de imagens de uso genérico são mostrados na Figura 1. O diagrama permite representar desde sistemas de baixo custo até sofisticadas estações de trabalho utilizadas em aplicações que envolvem uso de imagens. Abrange as principais etapas que se pode efetuar sobre uma imagem: aquisição, armazenamento, processamento e exibição. Além

disso, uma imagem pode ser transmitida à distância utilizando meios de comunicação disponíveis.

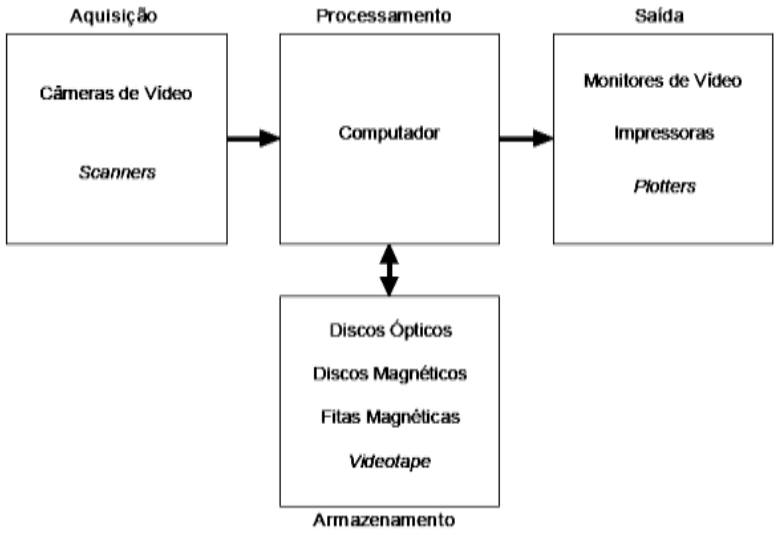

Figura 1. Elementos básicos de um sistema de processamento de imagens.
Fonte: Marques (1999).

4.3. Computação gráfica: História e evolução

A computação gráfica segundo Marques (1999), tem como propósito, pesquisar e desenvolver técnicas computacionais de transformação de dados em imagens e vice-versa. A importância da computação gráfica estendeu-se para os setores da economia moderna. Entretanto a multibilionária indústria do entretenimento, os setores de pesquisa científica de ponta, a indústria aeroespacial, o setor automobilístico, são apenas alguns exemplos de setores da economia que se beneficiam e dependem expressivamente do desenvolvimento da tecnologia da computação gráfica.

A revolução do computador pessoal só foi possível nas proporções observadas devido justamente ao desenvolvimento da computação gráfica. Esta evolução teve início com o projeto *Whirlwind*, um computador construído pelo MIT no final da década de 50 que foi considerado o primeiro computador digital capaz de operar em tempo real. Até então os computadores existentes funcionavam por meio de processamento de arquivos em lotes, isto é, o processamento dos dados ocorria sempre de forma seqüencial, com a entrada dos dados, o processamento e a saída dos resultados ocorrendo em fases distintas e sem possibilidade de interferência no processamento após a entrada dos dados.

O *Whirlwind* possuía uma interface de saída através de um tubo de raios catódicos (SEVO, 2005).

A computação gráfica evoluiu rapidamente, e no final da década de 60 os primeiros avanços no campo de modelagem 3D por computador foram obtidos. Um dos primeiros dispositivos de visualização tridimensional de que se tem notícia foi justamente um dispositivo estereoscópico, um *Head Mounted Display* (HMD) desenvolvido por Ivan Sutherland (SUTHERLAND, 1969), o mesmo pesquisador do MIT que desenvolvera o *sketchpad* alguns anos antes. Enquanto o *sketchpad* permitia ao seu usuário desenhar formas simples diretamente na tela do computador utilizando para isso uma caneta óptica, o HMD desenvolvido por Sutherland exibia duas imagens em *wireframe* separadas, uma para cada olho, criando assim uma ilusão tridimensional estereoscópica (SEVO, 2005).

Segundo Marques (1999) o advento do microprocessador no início da década de 70, a computação gráfica evolui para um patamar significativo. Em 1974 Ed Catmuli obteve o título de PhD em ciência da computação por sua tese sobre mapeamento de texturas, *Zbuffer* e superfícies curvas. Um avanço importante ocorrido na década de 80 foi a introdução da utilização de pequenos trechos de código denominados *shaders*, que geram texturas através de fórmulas matemáticas, em contraste com o mapeamento de texturas baseado em imagens 2D existente até então. Além disso, os *shaders* podem ser facilmente separados para execução em processadores separados da unidade central de processamento (CPU) de um sistema, uma característica que seria integralmente explorada alguns anos mais tarde com o advento das placas gráficas.

Os anos 90 foram a década das produções cinematográficas que exploravam intensivamente os efeitos especiais criados por meio do computador. Filmes como Jurassic Park (1994), Toy Story (1995), O Máscara (1995), para citar alguns, obtiveram sucesso e impressionaram por seus efeitos visuais, impressionantes à época. Este período também marcou a ascensão do mercado de entretenimento eletrônico, com o explosão do mercado de placas gráficas para computadores de vídeo games como o *Playstation 1* (SEVO, 2005).

Enquanto na décadas de 60 a 80 foram precussoras da criação de parte dos conceitos e técnicas computacionais, as duas últimas décadas foram a aplicação e extensão destas técnicas, graças ao aumento na potência e capacidade dos computadores. O realismo que vem sendo atingido pelos novos desenvolvimentos na área torna cada vez

mais difícil distinguir o que é uma imagem real de uma imagem gerada cem por cento por computador (SEVO, 2005).

4.4. Aquisição de imagens

A visão computacional existe em analogia ao sistema visual humano, e seguindo esta linha de raciocínio é recomendado um estudo da capacidade humana em visualizar e entender o cenário e o contexto ao seu redor. Os olhos atuam como sensores, onde a imagem é projetada e transformada em impulsos elétricos, e a informação proveniente destes chega ao cérebro para ser interpretada (SEVO, 2005).

A etapa de aquisição tem como função converter uma imagem em uma representação numérica adequada para o processamento digital subsequente. Este processo é composto por dois elementos principais. O primeiro é um dispositivo físico sensível a uma faixa de energia no espectro eletromagnético (como raio X, ultravioleta, espectro visível ou raios infravermelhos), que produz na saída um sinal elétrico proporcional ao nível de energia detectado. O segundo é o digitalizador propriamente dito que converte o sinal elétrico analógico em informação digital, isto é, que pode ser representada através de *bits* zeros e uns (0s e 1s) segundo Marques (1999).

Esteves (2011) demonstra na Figura 2 um diagrama esquemático de um olho ou globo ocular de um ser humano com suas subestruturas internas, dentre elas tem a retina, onde a luz é captada e convertida em sinal nervoso, e o cristalino, onde a imagem é focalizada. O sensoriamento ocular consiste na atuação de dois tipos de células fundamentais: os bastonetes, responsáveis por captar a luminosidade; e os cones, capazes de diferenciar as diversas cores existentes na imagem.

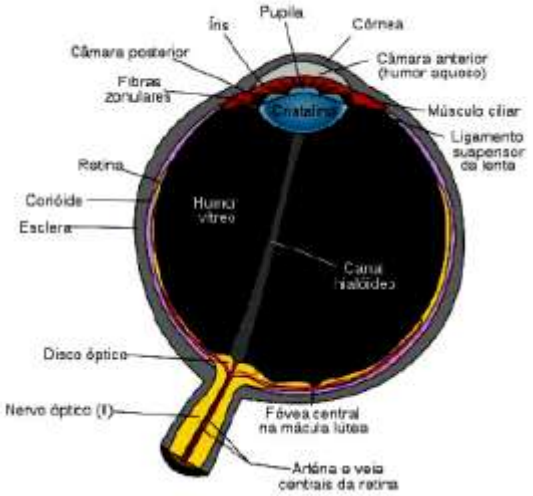

Figura 2. Diagrama esquemático do globo ocular humano.
Fonte: Esteves (2011).

Na Figura 3 observa-se o diagrama esquemático de uma estrutura composta por estas duas células. Existem várias dessas estruturas espalhadas homogeneamente pela retina, onde todas as informações são preparadas para serem enviadas ao cérebro por meio do nervo óptico (ESTEVES, 2011).

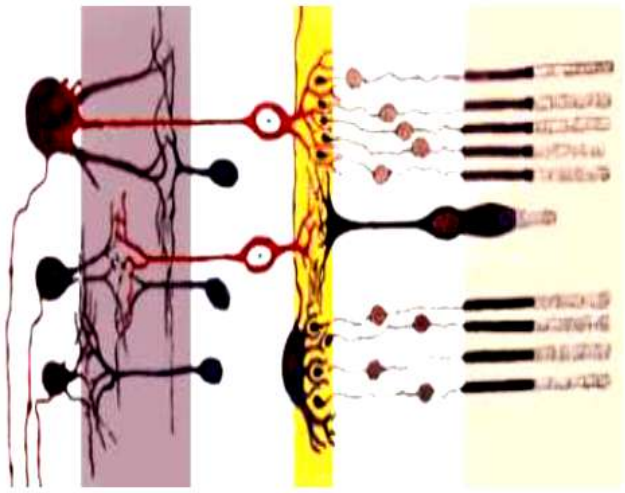

Figura 3. Perspertiva microscópica da retina humana.
Fonte: Esteves (2011).

Quando se trata de imagens digitais, o contexto é similar. A maioria das câmeras digitais funciona por meio de sensores retangulares dispostos lado a lado. O sensor mais usado é o dispositivo de carga acoplada (CCD – *charge coupled device*).

Outra tecnologia usada é a de semicondutor metal-óxido complementar (CMOS - *complementary metal oxide semicondutor*). Esses sensores tem a capacidade de converter a energia luminosa em carga elétrica e quando milhares deles são agrupados em uma matriz são capazes de capturar uma imagem digital. E assim, cada microssensor captura um elemento mínimo da imagem (*pixel picture element*) (RAPOSO, 2004).

Segundo Esteves (2011), a imagem digital consiste na informação capturada pelos sensores, onde cada um deles responde, ao mesmo tempo, um valor que corresponde à carga acumulada de energia luminosa incidida. Este valor irá variar de acordo com o brilho (intensidade) e a cor (frequência) da luz capturada. Em um CCD, a carga é conduzida através do circuito eletrônico e armazenada para, posteriormente, a sua leitura. Um conversor analógico-digital relaciona cada valor de quantidade de carga obtido a um valor digital ou binário. A Figura 4 mostra um sensor CCD.

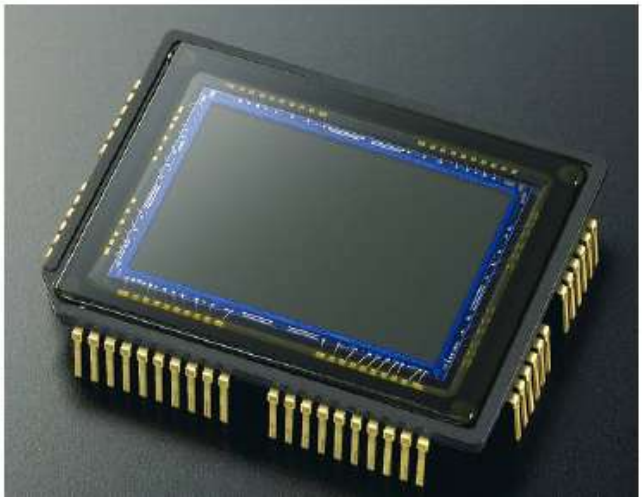

Figura 4. Sensor CCD.
Fonte: Esteves (2011).

Enquanto a Figura 5 apresenta um CCD junto a um circuito de uma câmera digital (ESTEVES, 2011).

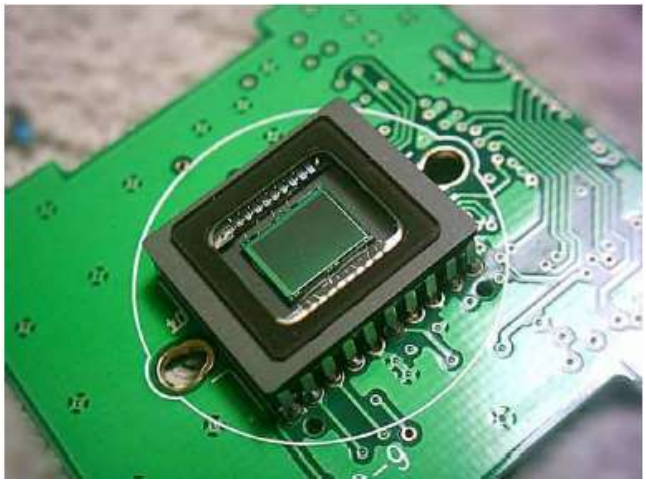

Figura 5. Sensor CCD junto a um circuito de uma câmera digital.
Fonte: Esteves (2011).

Para não necessitar de um conversor analógico digital, os dispositivos CMOS consistem de uma matriz de pequenos sensores, onde cada um destes possui micro transistores que fazem a leitura da carga diretamente para valores digitais. Por não precisar de um conversor, o tempo de resposta deste dispositivo é bem menor quando comparado ao CCD e o seu consumo de energia elétrica também menor. Porém a informação é mais suscetível a defeitos "ruidos", já que cada sensor vai ter uma leitura individual, ocorrendo assim aleatoriedade nos erros de captação de cada elemento ou pixel da imagem. A Figura 6 demonstra um sensor CMOS (ESTEVES, 2011).

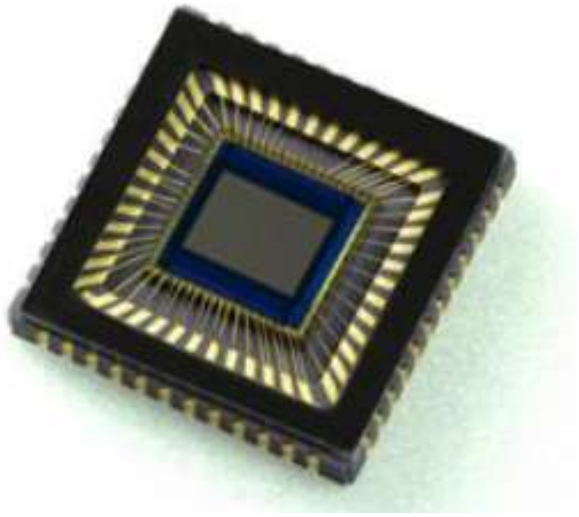

Figura 6. O sensor CMOS.
Fonte: Esteves (2011).

Quanto mais sensores existirem em um dispositivo, maior será a resolução da câmera. A Figura 7 apresenta as diferentes resoluções possíveis para uma

mesma imagem. O valor de 1 *megapixel* representa a quantidade de elementos ou *pixels* da imagem digital, ou seja, 1 milhão de *pixels*. Pode se dizer também que este valor corresponde à área da imagem, já que uma imagem digital com 1600 *pixels* de comprimento e 900 *pixels* de altura tem aproximadamente 1,4 *megapixels* (1600 x 900 = 1440000) (GONZALEZ, 2000).

Figura 7. As diversas resoluções para uma mesma imagem em medição de *Megapixel.*
Fonte: Gonzalez (2000).

4.5. Captação de cores

Os sensores das câmeras digitais segundo Gonzalez (2000), CCD e o CMOS, convertem a intensidade de luz para um valor que possa ser, posteriormente, processado. Esta funcionalidade é similar a dos bastonetes da retina do olho humano, porém só é possível enxergar em preto e branco. Para ser possível a aquisição ou captação de cores, o ser humano tem os cones na retina dos olhos. Então, para existir esta funcionalidade em uma câmera, é necessário usar um artifício que utiliza a ideia de dividir ou filtrar a captação em três cores primárias para cada elemento ou pixel da imagem. Estes três valores resultantes são combinados para obter-se uma cor de espectro.

Para cada *pixel* deverá existir três sensores, e cada um destes terá um filtro para determinada faixa de cor primária. Neste caso, terá que existir um divisor de feixe que terá a função de dividir por igual a intensidade de luz para cada um dos três sensores conforme é demonstrado na Figura 8. Dessa forma, os dispositivos de

sensoriamento, que usam essa ideia, tem melhor qualidade, contudo são mais densos e mais caros (GONZALES, 2000).

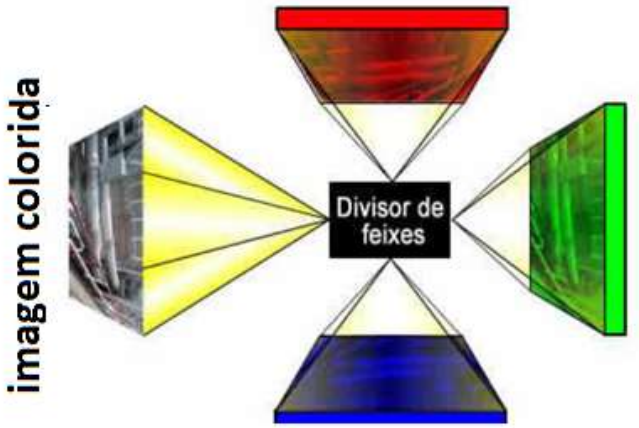

Figura 8. As Imagem colorida resultante.
Fonte: Gonzalez (2000).

Segundo Gonzalez (2000), outra possibilidade para captação de cores, e também a mais comum, é utilizar a interpolação de *pixels*. Neste método, cada elemento terá um único sensor com apenas um filtro de determinada cor primária, interpolando ou alternado as cores dos filtros ao longo da imagem digital. A cor do espectro de cada *pixel* é encontrada por meio de uma estimativa entre o valor adquirido no sensor correspondente e nos seus vizinhos. O filtro de Bayer, apresentado na Figura 9, consiste na ideia de enfileirar alternadamente filtros vermelhos e verdes, e filtros azuis e verdes, mudando de um padrão a outro para cada linha horizontal ou vertical da imagem. Uma maior quantidade do filtro de verde explica-se pelo fato de ser a cor que o olho humano é mais sensível, assim a imagem digital adquirida terá mais valor significativo em sua coloração (ESTEVES, 2011).

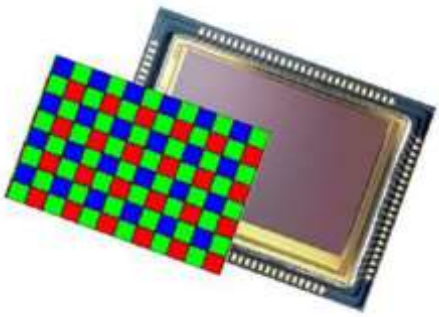

Figura 9. O filtro de Bayer em um sensor de câmera digital.
Fonte: Esteves (2011).

4.6. Modelos de câmeras

A imagem digital é adquirida através da projeção de uma imagem real em um plano imaginário, chamado de plano de imagem da câmera, onde se encontram os sensores. Essa projeção tem como resultado uma matriz a qual se dá o nome de matriz de projeção da câmera. Cada câmera tem sua matriz de projeção característica, sendo definida pelo seu tipo de modelo e seu processo de fabricação. A variação da matriz de projeção de uma câmera está diretamente relacionada à distância focal desta. Existem outros fatores, tais como variações térmicas ou mecânicas, que podem causar alterações nessa matriz. A regulagem da distância focal é um recurso necessário nas câmeras, pois essa distância da lente entre o sensor da câmera pode detalhar a imagem digital dependendo da profundidade ou distância entre os pontos dispostos na imagem real e a lente (GONZALEZ, 2000).

Segundo Gonzalez (2000) isso acontece porque a imagem tende a corresponder uma vista ampla quando a lente é aproximada do sensor, enquanto o aumento da distância focal torna menor a área captada pelo sensor ocorrendo assim um aumento centralizado na imagem adquirida Figura 10.

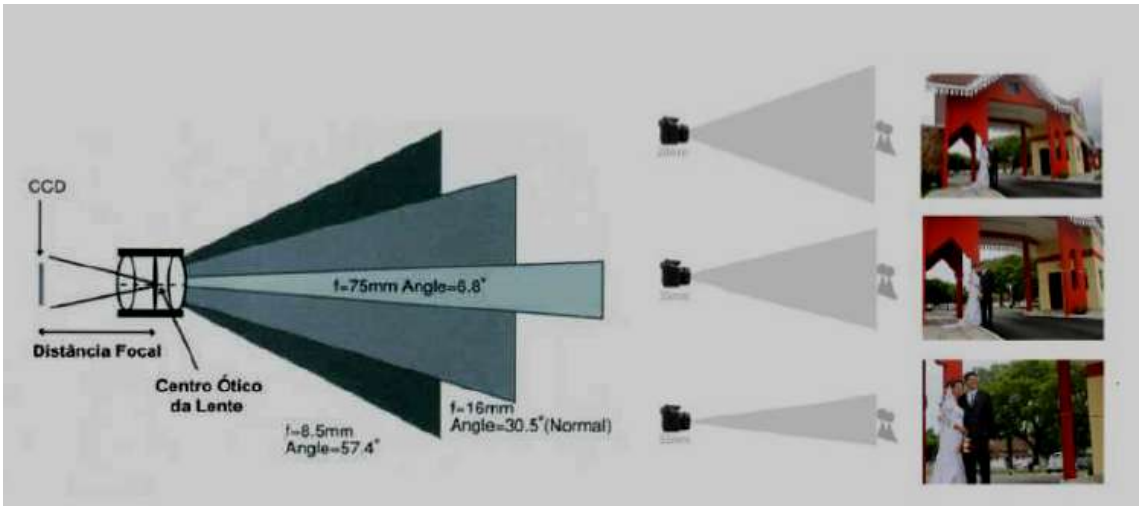

Figura 10. Modificação da distância focal para obter aumento centralizado de uma mesma imagem.
Fonte: Gonzalez (2000).

Segundo Gonzalez (2000), dentre os principais tipos de câmeras, destaca-se: as de foco fixo, as de foco automático, e as pinhole. A diferença entre esses tipos é a maneira em que a imagem se projeta no sensor da câmera. No modelo pinhole, a luz passa por um caminho estreito antes de ser captada, conforme Figura 11.

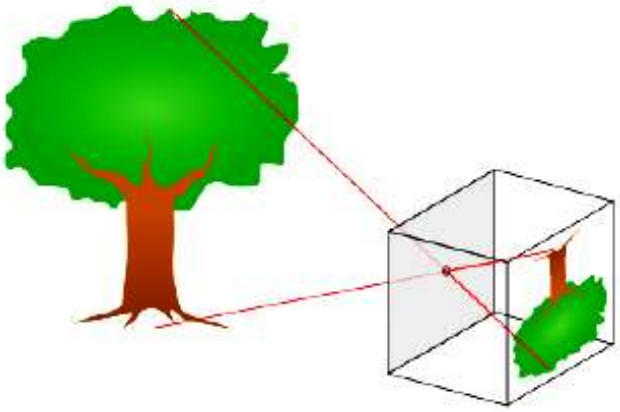

Figura 11. Modificação captura de uma imagem em uma câmera *pinhole*.
Fonte: Gonzalez (2000).

Para existir uma regulagem na distância focal da imagem projetada e evitar distorções, a luz é desviada por lentes convergentes antes de atingir o sensor Figura 12.

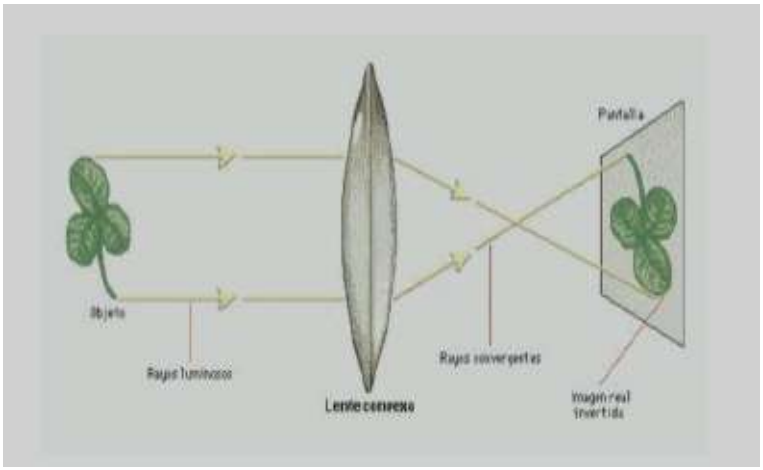

Figura 12. Captura de uma imagem após a luz passar por uma lente convexa.
Fonte: Gonzalez (2000).

Segundo Esteves (2011), o plano de imagem da câmera tem tamanho fixo, o ângulo formado entre os pontos da extremidade da imagem adquirida e o ponto de passagem da luz é diretamente relacionado à distância focal conforme também mostra a Figura 10. Esse ângulo é chamado de ângulo de convergência da imagem e é de fundamental importância na calibragem das câmeras usadas nas aplicações de visão computacional. Conforme um objeto se movimenta paralelamente em relação ao plano de imagem da câmera, o deslocamento será correspondido por uma mudança de posição de todos os pontos deste objeto imagem adquirida. Quando este se movimenta perpendicular

em relação ao plano de imagem da câmera haverá uma ampliação ou redução do seu tamanho no sensor, e é essa peculiaridade associada à noção de tamanho relativo que torna possível o cálculo de profundidade em uma visão monocular Figura 13.

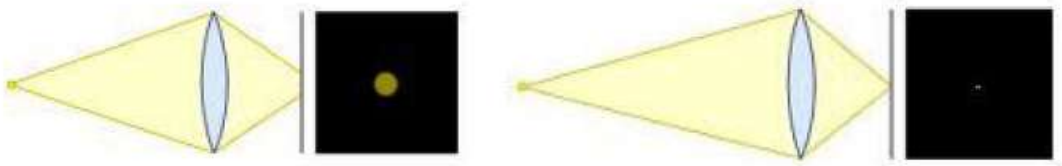

Figura 13. Afastamento de um objeto e seu efeito de redução de tamanho na imagem capturada
Fonte: Esteves (2011).

Essa ampliação ou redução ocorre por causa da convergência da luz emitida pela imagem real até o sensor. Os feixes de luz se aproximam ou se afastam do centro da imagem do objeto projetado no plano de imagem da câmera Figura 14 (ESTEVES, 2011).

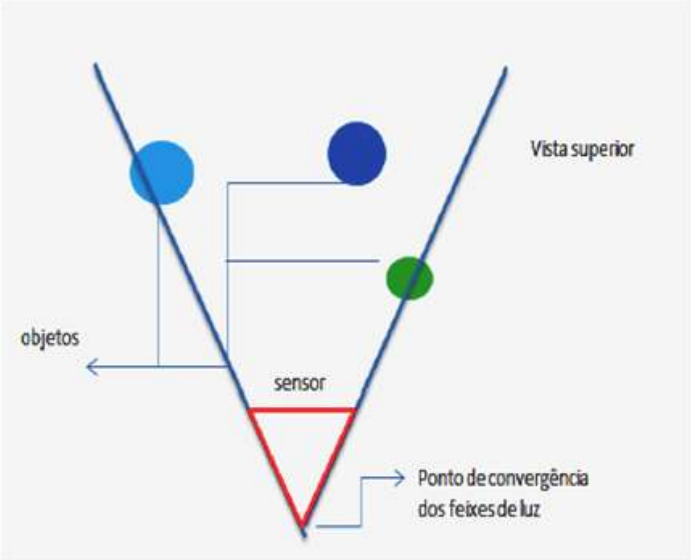

Figura 14. Esquema de uma captação vista de cima, de objetos em posições diferentes
Fonte: Esteves (2011).

4.7. Conceito de imagem digital

Uma imagem digital é uma matriz bidimensional de *pixels*. Cada *pixel* de uma imagem é independente e pode representar qualquer informação visual, ou seja, uma imagem pode ser uma matriz completamente aleatória (RAPOSO, 2004).

Segundo Gonzalez(2000), uma imagem digital é uma função f(x,y) discretizada tanto em coordenadas espaciais quanto em brilho. As coordenadas x e y são positivas, pelo conceito matemático de que uma matriz não tem índice negativo, e a origem parte do ponto superior esquerdo da imagem Figura 15.

Imagem é binarizada quando, cada *pixel* que a compõe só possa representar duas informações ou cores diferentes, a cor preta e a cor branca. Na maioria das API's (*Application Programming Interface*) ferramentas de programação que aborda processamento de imagens digitas, no que se refere às imagens binarizadas, a cor preta é representada pelo valor zero, e a cor branca é atribuído o valor 1. Uma imagem binarizada pode ser resultado da extração de alguma característica ou informação de uma imagem colorida ou de tons de cinza. Geralmente, após a filtragem ou extração de características de uma imagem, é necessário utilizar uma técnica de limiarização para segmentar uma imagem em dois grupos distintos, tendo como resultado uma imagem binarizada (RAPOSO, 2004).

A técnica de limiarização consiste em agrupar níveis de cinza (considera-se uma faixa de 0 a 255, em que 0 representa a cor preta e 255 a cor branca) de uma imagem de acordo com a definição de um limiar. No caso mais simples, um único limiar é definido. Nessa técnica, todos os *pixels* do histograma da imagem são rotulados como sendo do objeto ou do *background.* A técnica de limiarização é aplicada no exemplo da Figura 15 (RAPOSO, 2004).

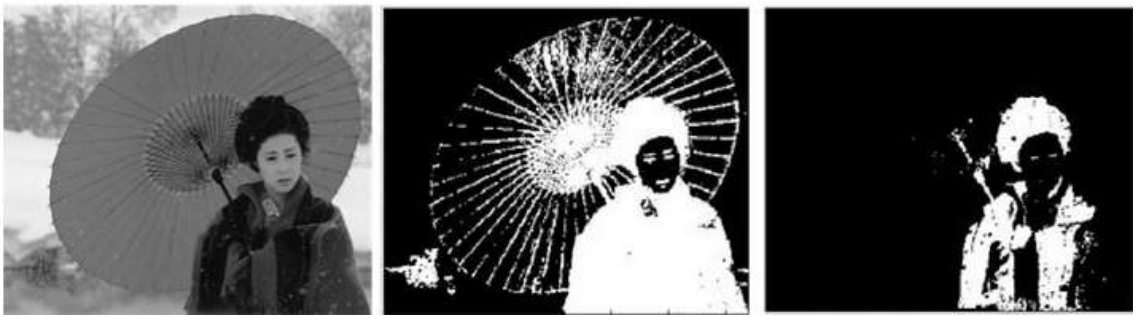

Figura 15. Da esquerda para a direita, a imagem original, seguida pela mesma imagem aplicando-se um limiar 30 e 10, respectivamente.
Fonte: Raposo (2004).

Conforme (GONZALEZ, 2000), em uma imagem de tons de cinza, como a imagem da Figura 16(a), uma das representações é a sua matriz possui valores decimais de 0 à 255, conforme a Figura 16(c) mostra, representados por números binários,

ou seja, 1 byte. As diferentes intensidades, partindo do escuro em direção ao claro, são representadas com um valor crescente conforme mostra a Figura 16(b), em que o valor 255 representa a cor branca e o zero representa a cor preta.

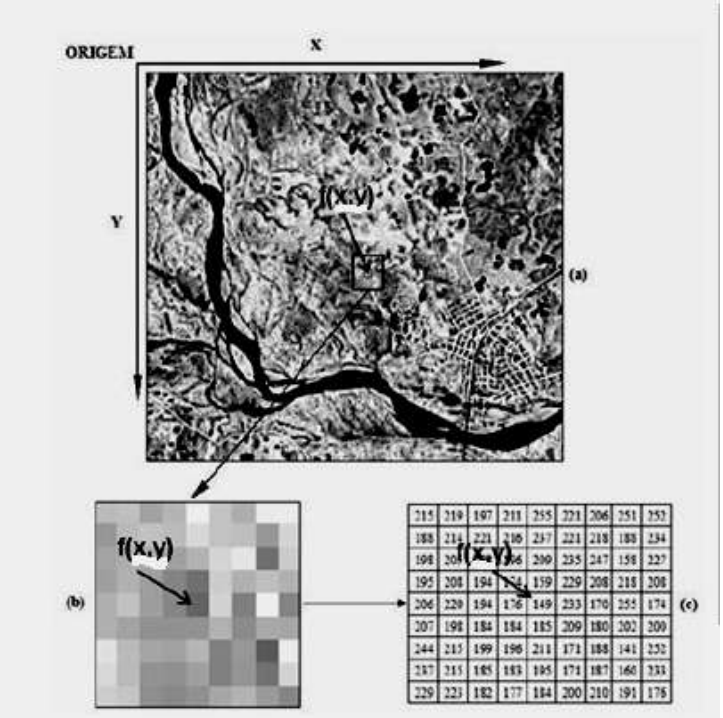

Figura 16. (a) Imagem monocromática, (b) Uma visão aproximada de uma pequena área da imagem, (c) Valores de cada pixel da imagem.
Fonte: Gonzalez (2000).

Aumentando a quantidade de informação, as imagens coloridas necessitam de valores relacionados à cor de espectro de cada pixel. Dentre os modelos de representação existem o ciano, magenta e amarelo (CMY – *cyan, magenta, yellow*), e o vermelho, verde, azul (RGB – *red, green, blue*). No modelo RGB, a cor de um *pixel* é representada pelas três cores primárias que dão nome ao mesmo. A Figura 17 mostra algumas cores com seus valores que correspondem ao modelo RGB (GONZALEZ , 2000).

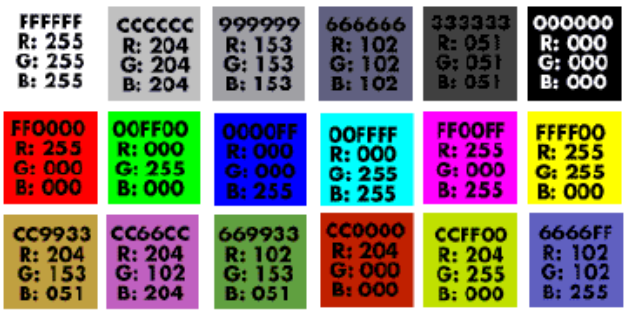

Figura 17. Valores em hexadecimal e decimal de algumas cores no modelo RGB.
Fonte: Gonzalez (2000).

Segundo Gonzalez (2000), estes modelos são funcionais porque os três sinais emitidos pelos *pixels*, quando muito próximos, são visualmente sobrepostos causando assim um somatório dos comprimentos de ondas. Então um sinal vai interferir nos demais, e assim uma nova cor será mostrada por essa mistura aditiva. Pode ser útil decompor uma imagem em cores primárias para facilitar a identificação de elementos ou pontos de interesse, ou então para ter a finalidade de simplificar ou diminuir a quantidade de informação a ser posteriormente analisada. A Figura 18 mostra o modelo RGB aplicado a uma imagem colorida usando esse modelo.

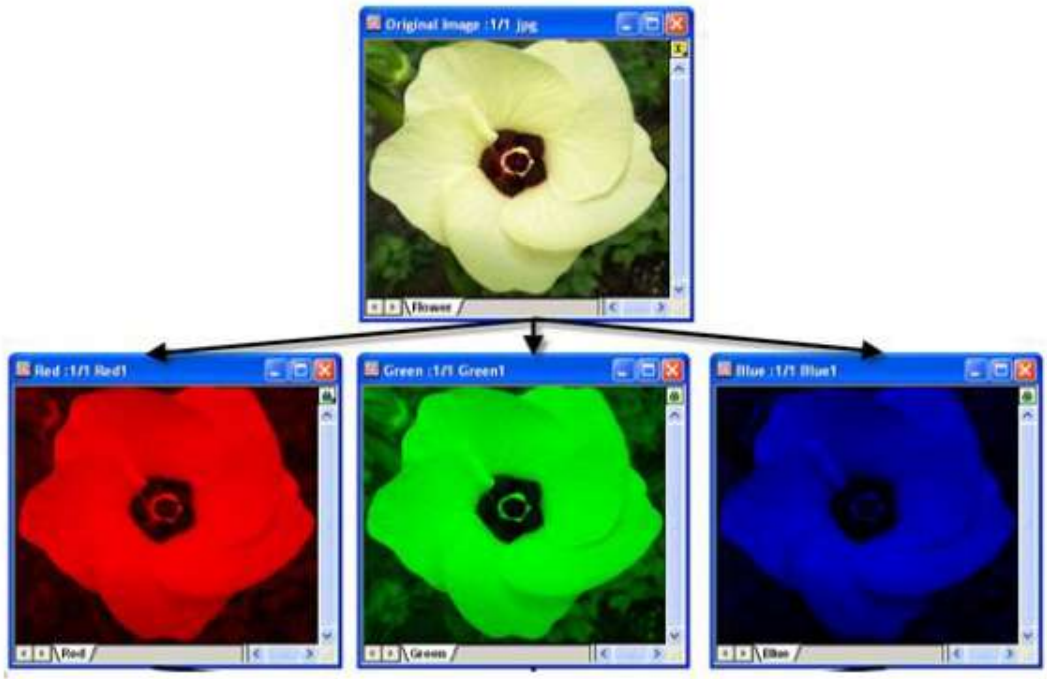

Figura 18. Imagem decomposta em 3 componentes.
Fonte: Gonzalez (2000).

4.8. Estereoscopia

Bertozzi e Broggia (1997) estereoscopia (do grego *stereós* = sólido, firme; do latim *copia* = traslado, reprodução) é uma técnica que visa analisar duas ou mais imagens digitais obtidas em perspectivas diferentes, e assim obter informações do espaço tridimensional. O sistema visual humano é binocular e utiliza essa técnica naturalmente por meio dos olhos, que capturam o par estereoscópico, e do cérebro, que processa essas imagens bidimensionais e as conceitua em informações tridimensionais.

Segundo (GONZALEZ, 2000), no momento em que uma foto é capturada por uma câmera digital, a imagem resultante é uma representação bidimensional de um espaço tridimensional. Logicamente, alguns fatores dessa imagem podem causar a percepção de profundidade, tais como: iluminação, perspectiva, oclusão, sombra, gradiente da textura, dentre outros. A perspectiva se baseia na ideia de que o tamanho dos objetos da imagem digital diminui à medida que o seu correspondente real se afasta do ponto de captura. As distâncias entre os objetos também diminuem à medida que estes se afastam. A iluminação pode causar a noção de profundidade através da reflexão. A variação de iluminação ao longo de uma superfície fornece informações de formato e de curvatura. A oclusão ou interposição ou interrupção de contorno acontece quando um objeto se encontra a frente de outro, ocultando informações relativas a este. A sombra adiciona a informação de posição relativa de um objeto a outro na imagem digital, pois por meio disso é possível saber se os objetos estão encostados ou afastados, ou se um deles está à frente do outro e vice-versa.

O gradiente da textura se refere a características ou padrões que podem classificar e distinguir as diversas regiões ou conjunto de elementos da imagem digital. Assim, quanto mais distante o objeto, mais densa fica sua textura e, por fim, menos nítida. Para existir a visão estereoscópica, é necessário poder extrair informações de duas ou mais perspectivas diferentes. O sistema visual humano é binocular e assim o cérebro recebe duas perspectivas do que é visto. Então um olho vê uma imagem ligeiramente diferente do outro, onde a diferença é compatível à profundidade de cada ponto da imagem. Essa diferença ou deslocamento entre as imagens recebe o nome de disparidade. A disparidade entre duas imagens capturadas em perspectivas diferentes, por câmeras digitais, pode ser usada no cálculo para achar a distância de um determinado objeto ou marcador. E quanto mais perto o objeto estiver das câmeras, maior será a disparidade. Existem duas abordagens na estereoscopia: a triangulação e o pareamento de pontos do par estereoscópico (GONZALEZ, 2000).

4.8.1. Princípios de triangulação

Segundo (GONZALEZ, 2000), a estereoscopia se trata da aplicação de vários cálculos de triangulação para cada um dos diferentes pontos pareados por duas imagens. A distância do objeto P em relação ao segmento EC, onde os pontos E e C

correspondem respectivamente às câmeras esquerda e direita, é representada por D. O ângulo α é formado através do encontro dos segmentos PE e EC, e o ângulo β corresponde ao encontro dos segmentos PC e EC. A Figura 19 é uma visualização desse cenário.

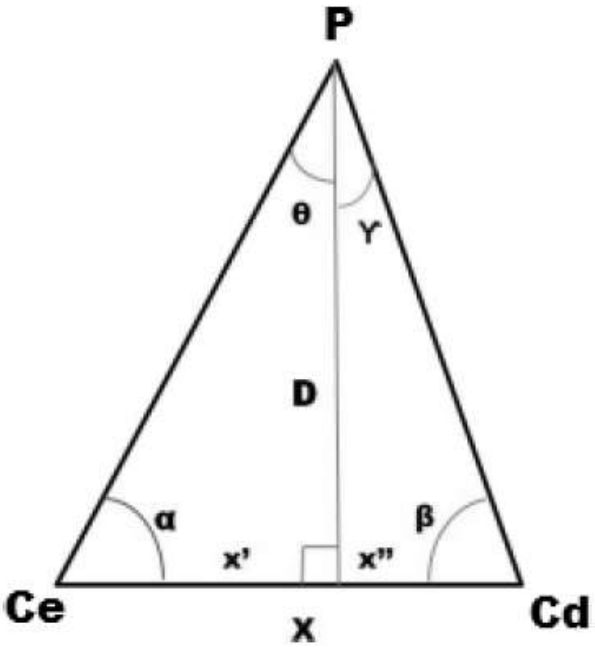

Figura 19. Triangulação para encontrar a distância D.
Fonte: Gonzalez (2000).

Lei dos Senos é formulada pelas equações:

1) $D/\text{sen}(\beta) = x''/\text{sen}(\gamma)$ $x'' = D.\text{sen}(\gamma)/\text{sen}(\beta)$,

2) $D/\text{sen}(\alpha) = x''/\text{sen}(\theta)$ $x'' = D.\text{sen}(\theta)/\text{sen}(\alpha)$,

3) $X = x' + x'' = (D.\text{sen}(\gamma)/\text{sen}(\beta))+(D.\text{sen}(\theta)/\text{sen}(\alpha))$.

A soma dos ângulos internos de um triângulo é 180:

4) $\alpha + \theta + 90° = 180°$ $\theta = 90° - \alpha$, (4)

5) $\beta + \gamma + 90° = 180°$ $\gamma = 90° - \beta$. (5)

A partir das equações 3, 4 e 5 é possível obter a seguinte expressão:

6) $X = (D.\text{sen}(90°- \alpha)/\text{sen}(\alpha)) + (D.\text{sen}(90°- \beta)/\text{sen}(\beta))$.

E utilizando a relação seno-cosseno,

7) $\text{sen}(90°- \omega) = \cos(\omega)$,

na equação 6 se obtém:

8) $X = (D.\cos(\alpha)/\text{sen}(\alpha)) + (D.\cos(\beta)/\text{sen}(\beta))$.

Usando a relação trigonométrica,

9) $\cos(\omega)/\text{sen}(\omega) = 1/\text{tg}(\omega)$,

na equação 8 é possível obter:

10) X = (D/tg(α)) + (D/tg(β)).

Colocando em evidência o valor de D:

11) D=X / [(1/tg(α)) + (1/tg(β))].

Por meio dessa equação (11), é possível encontrar a distância de um objeto em relação ao observador apenas conhecendo os ângulos formados entre os dois pontos de observação e o objeto. Porém, numa imagem digital as informações relativas aos ângulos de projeção não são diretamente obtidos. Então se faz necessário algumas adaptações para poder utilizar a fórmula de triangulação em um par de imagens estereoscópicas. A Figura 20 corresponde à situação visualizada na Figura 19 em conjunto com o modelo de câmera Figura 14 (GONZALEZ, 2000).

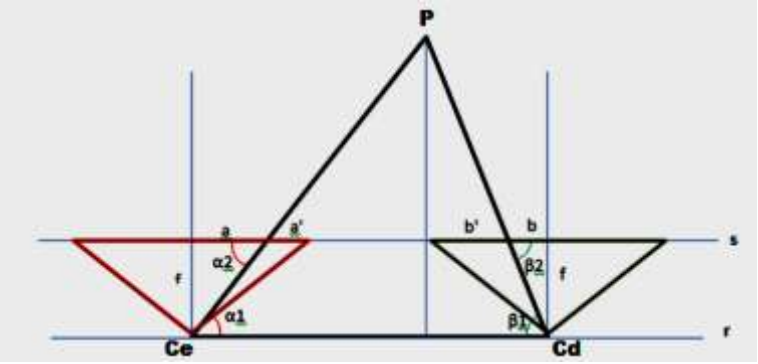

Figura 20. Visão estereoscópica em conjunto com a ideia de ângulo de convergência e distância focal.
Fonte: Gonzalez (2000).

Sabendo-se que a reta r é paralela a reta s, o ângulo α1 possui grau de abertura igual ao ângulo α2 por que são alternos internos entre si, assim como os ângulos β1 e β2.

Então o valor da tangente de α1 é obtida pela fórmula:

12) tg(α) = f/a,

e da mesma forma:

13) tg(β) = f/b. (13)

Incluindo essa ideia no cálculo da distância:

14) D = X/ [(1/f/a) + (1/f/b)] = X/[(a/f)+(b/f)] = X/[(1/f).(a+b)] = f.X/(a+b).

Onde os valores de a e b podem ser obtidas pelas imagens digitais capturadas em unidades de *pixels*. Os valores referem-se à distância em *pixels* do ponto da imagem até o eixo vertical localizado no centro da imagem Figura 15.

É possível uma simplificação ao usar o valor absoluto dos pontos a e b, ou seja a coordenada horizontal, na equação 14, que serão representados por a' e b', respectivamente. Sabendo-se do valor total da largura da imagem em *pixels* I e que este valor é igual nas duas imagens (câmera esquerda e direita), no caso de serem utilizadas câmeras de modelo igual, o valor de a será substituído por I/2 – a' e o valor de b será substituído por I/2 – b' resultando em:

$$D = f.X/[(I/2)-a'+(I/2)-b'] = f.X/(I - a' - b') \quad (15)$$

O termo I – a' representa a distância do ponto até a lateral esquerda da imagem da câmera direita em *pixels*, assim como o termo b' representa a distância do ponto até a margem esquerda da imagem da câmera esquerda. Para simplificar mais ainda a fórmula, I – a' será substituído por Xd e b' será substituído por Xe e isto é apresentado na equação a seguir:

$$D = f.X/(Xd - Xe) \quad (16)$$

Segundo Gonzalez (2000), f é o valor da distância focal das duas câmeras (valor igual para as duas câmeras, quando do mesmo modelo) que permanecerá fixo durante todo o processo da aplicação e o valor de X é ajustável (afastamento das câmeras) e influencia na precisão do cálculo da distância: quando o valor de X for alto, a precisão é melhor para calcular longas distâncias. Assim o valor de fX é a máxima distância possível a ser calculada. Por exemplo, para um valor fX igual a 1 metro.*pixel* o cálculo não corresponderá a distância maiores que 1 metro, porque não existe variação ou deslocamento menor que 1 *pixel*. Então não é conveniente calcular a distância perto do limite, pois a variação de 1 para 2 *pixels* causa uma diferença de metade da distância (no caso do exemplo anterior, a variação de 1 *pixel* corresponde a 1 metro e a variação de 2 *pixels* resulta em 50 centímetros, ou seja, uma variação de 50% no cálculo da distância).

A Figura 16, mostra a variação ou deslocamento de *pixels* em relação à distância. É possível notar que enquanto a variação dos *pixels* for alta mais precisa será a medição.

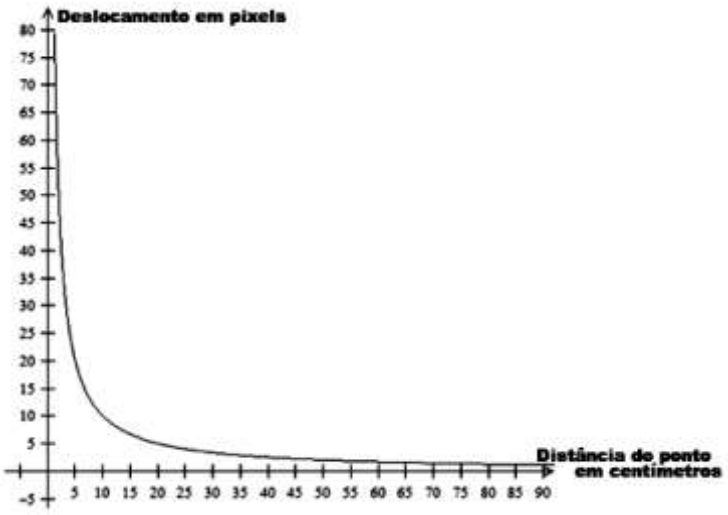

Figura 21. Perda de precisão do deslocamento y das imagens conforme se aumenta a distância x dos pontos a serem emparelhados com a estereoscopia.
Fonte: Gonzalez (2000).

Um fator importante na precisão do cálculo da distância é a resolução das câmeras digitais. Uma câmera com alta resolução será mais precisa, pois terá uma variação ou deslocamento de uma maior quantidade de *pixels*. Por exemplo, um par de câmeras digitais capturando imagens com 1000 *pixels* de comprimento terá o fator Xd-Xe variando de 0 até 999. A variação de 0 *pixels* ocorre quando a distância está tão alta que não causa deslocamento nas duas imagens e a variação 999 *pixels* acontece quando a distância está tão próxima que o ponto se desloca da borda esquerda da câmera direita para a borda direita da câmera esquerda.

Assim um par de câmeras digitais com resolução de 1600 *pixels* horizontais tem melhor precisão quando comparado a um com resolução de 800 *pixels*, pois a primeira tem um maior deslocamento em *pixels* por proximidade.

4.8.2. Pareamento dos pontos

O cálculo da distância simplificado consiste de operações matemáticas de subtração e divisão, sem comparações ou iterações, e assim é uma tarefa de custo computacional desprezível. Outra tarefa que dificulta o cálculo da distância é o pareamento dos pontos do par estereoscópico de imagens. O uso de medidas de similaridade, tais como Minkowski, Hamming, Tanimoto e Entropia Cruzada, para parear os pontos pode se tornar uma tarefa complexa (alto custo computacional) e passível de

erros (falsos pares). Porém, para um objeto ou uma marca de uma cor ou uma característica única, essa tarefa se torna bastante simplificada (GONZALEZ, 2000).

O cenário onde o par de câmeras digitais captura duas imagens com diversos objetos sem nenhum padrão e espalhados de maneira aleatória (sem uniformidade) o problema de pareamento se torna mais complexo. Além de que, existirão pontos que uma câmera captará e a outra não, por causa do ângulo de convergência, e assim haverá um aumento do número de informação irrelevante para o pareamento dos pontos.

Para simplificar as medidas de similaridade existe a possibilidade de filtrar a informação da imagem por determinadas faixas de cor (delimitar as faixas manualmente ou por *clustering*) ou por variação de *pixels* vizinhos da imagem. Neste último caso, os pontos seriam as bordas de algum objeto, ou então as linhas da imagem.

Uma abordagem consiste em encontrar um ponto em uma das imagens e depois procurá-lo na outra imagem. A outra se fundamenta em achar os pontos em ambas as imagens e depois tentar correlaciona-los. No caso da primeira abordagem, o custo computacional é elevado pois o espaço de busca é superdimensionado (em uma imagem de 10 *megapixels*, seriam 10 milhões de pontos em cada busca). Em outra abordagem, pode existir mais de um par para um único ponto (ambiguidade) e com isso podem vir a ocorrer erros, porém a complexidade ou o custo computacional é reduzido.

4.8.3. Técnicas de estereoscopia

As técnicas mais comuns para a produção artificial do efeito de estereoscopia por meio da classificação de equipamentos para a estereoscopia são:

Estéreo Passivo: As duas imagens são exibidas juntas e os óculos funcionam como filtros, separando cada imagem sobre o olho respectivo. Exemplo: anaglifo, polarização de luz.

Estéreo Ativo : Nesse tipo de sistema os óculos são capazes de exibir em cada olho a sua imagem correspondente com a ajuda do sistema de exibição do estéreo. Exemplo: óculos obturadores.

Independente do modo utilizado (ativo ou passivo), a separação das imagens para cada olho requer um dispositivo de visualização que variam desde simples

óculos com filtros até óculos com obturadores eletrônicos, utilizados nas técnicas mais sofisticadas (RAPOSO, et al. 2004).

4.8.4. Par estereoscópico

Segundo Gonzalez (2000), um par estereoscópico é um conjunto de duas imagens bidimensionais tomadas de uma mesma cena, com uma pequena separação horizontal entre os pontos de captura da imagem, simulando o espaçamento entre os olhos humanos. O princípio básico de todos os métodos de produção artificial de estereoscopia consiste em obter ou criar as duas imagens que compõem um par estereoscópico e apresentar separadamente cada imagem do par ao seu olho correspondente (direito ou esquerdo) Figura 22.

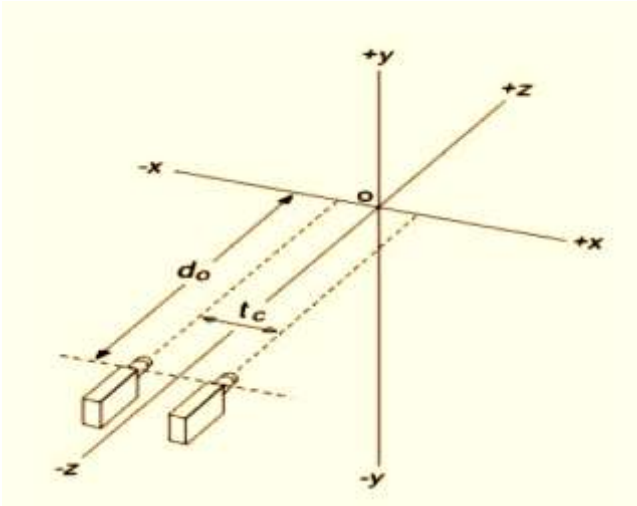

Figura 22. Obtenção de um par estereoscópico, separação horizontal entre as câmeras.
Fonte: Gonzalez (2000).

4.9. Anaglifos

A criação do efeito de estereoscopia segundo Albulquerque (2006), utilizando anaglifos é feita pela separação das imagens por cores. Cada imagem do par estéreo é composta por uma ou mais cores primárias de forma que a imagem esquerda não possua uma das cores da imagem direita e viceversa. Com este princípio é possivel separar as imagens esquerda e direita utilizando óculos com filtros coloridos, específicos para essa finalidade.

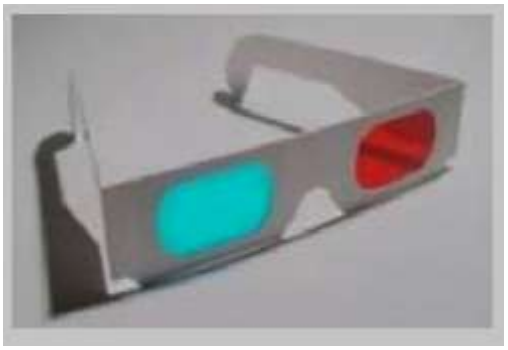

Figura 23. Óculos utilizados para visão estéreo com anaglifo.
Fonte: (TORI et al., 2006).

Segundo (RAPOSO, 2004), cada um dos olhos enxerga a imagem através de um filtro diferente, recebendo apenas uma das imagens do par estereoscópico. O filtro vermelho deixa atingir o olho apenas a imagem vermelha do anaglifo e o filtro azul e verde transmite ao olho apenas a imagem azul e verde. A Figura 24 ilustra essa separação de cores. A Figura 25 mostra um exemplo de anaglifo.

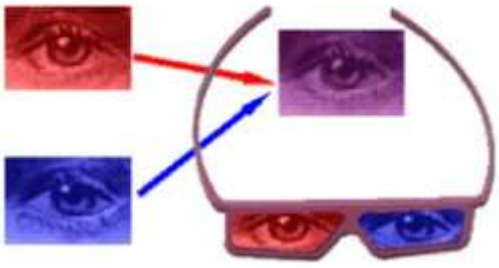

Figura 24. Separação de cores através dos óculos anaglíficos.
Fonte: (RAPOSO, et al. 2004).

Figura 25. Exemplo de anaglifo.
Fonte: (TORI et al., 2006).

Exibir um anaglifo requer apenas um projetor ou monitor e um par de óculos anaglíficos. O anaglifo pode ser impresso e é uma solução de baixo custo. A

principal desvantagem é a perda de qualidade na imagem devido à perda de cores, fator este que pode ser atenuado conforme será exposto adiante.

A implementação de estereoscopia acessível e de baixo custo por meio da técnica de anaglifos apresenta diversas vantagens interessantes que justificam a sua adoção, por exemplo:

1) Baixo custo de reprodução e visualização, podendo ser exibida pormonitores ou projetores comuns e visualizados através de óculos com filtros coloridos simples e de fácil confecção;

2) O efeito estéreo gerado pelo anaglifo independe da mídia de visualização, podendo ser impresso, projetado, estático, animado, etc;

3) A perda de qualidade causada pela coloração pode ser atenuada utilizando-se modos de anaglifo colorido, que conseguem uma reprodução parcial das cores da imagem original.

Devido a estes fatos, a estereoscopia utilizando anaglifos foi o método selecionado como modelo.

4.9.1. Métodos de obtenção de anaglifos

Segundo Gonzalez (2000), a obtenção de um anaglifo no computador consiste na obtenção das duas imagens do par estereoscópico e na coloração adequada de cada uma das imagens pixel a pixel, com a posterior soma dos *pixels* das duas imagens. Existem métodos diferentes para a obtenção de uma imagem em estéreo com anaglifo, e cada um apresenta características, vantagens e desvantagens.

4.9.2. Anaglifo puro

Segundo Gonzalez (2000), anaglifo puro consiste na remoção de todas as cores da imagem original, aplicando aos componentes vermelho e azul da imagem final quantidades proporcionais de cada componente RGB das imagens esquerda e direita, respectivamente. A representação da imagem pode ser descrita como:

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0{,}299 & 0{,}587 & 0{,}114 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

Este método possui as seguintes características:

Escurecimento da imagem;

Perda das cores;

Pouco vazamento de cor nos filtros.

O vazamento de cor ocorre quando a imagem de um olho não pode ser filtrada completamente pelo seu filtro correspondente, resultando em porções da imagem esquerda sendo enxergadas pelo olho direito, e vice-versa. Está associado à qualidade dos filtros utilizados e também à proporção dos canais verde e azul presentes na imagem esquerda (correspondente ao filtro vermelho), e vice-versa.

Figura 26. Anaglifo puro.

4.9.3. Anaglifo em escala de cinza

Segundo Gonzalez (2000), este método acrescenta à imagem final uma componente no canal verde, composta proporcionalmente pelas cores da imagem original direita, resultando em uma imagem final em escala de cinza. A representação da imagem pode ser descrita como:

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0{,}299 & 0{,}587 & 0{,}114 \\ 0{,}299 & 0{,}587 & 0{,}114 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

Este método apresenta as seguintes características:

Perda das cores, porém com a imagem em escala de cinza;

Mais vazamento do que em imagens com anaglifo puro.

Figura 27. Anaglifo em escala de cinza.

4.9.4. Anaglifo em cores

Segundo Gonzalez (2000), este método consiste na utilização integral das componentes vermelha, azul e verde da imagem original, sendo a componente vermelha tomada da imagem esquerda e as componentes azul e verde tomadas da imagem direita:

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

Utilizando a fórmula acima, obtemos um anaglifo com:

Reprodução parcial de cores;

Aparecimento de rivalidade binocular, ou rivalidade entre retinas.

A rivalidade binocular ocorre quando o cérebro se recusa a interpretar separadamente as informações recebidas em cada olho, fazendo com que as cores dos objetos pareçam saltar ou piscar durante a visualização da imagem. Isto ocorre devido à supressão alternada de cada imagem, ou seja, o cérebro considera apenas uma das imagens recebidas de cada vez. Eventualmente o olho dominante irá determinar qual

imagem será interpretada, porém este fenômeno pode provocar incômodo e fadiga durante a visualização (SOUSA, 2006).

Nos anaglifos, a rivalidade retínica ocorre em dois níveis: devido às diferenças nas cores captadas por cada olho, e devido às diferenças de brilho e contraste existentes entre as imagens esquerda e direita após a modificação das cores.

Figura 28. Anaglifo em cores.

4.9.5. Anaglifo otimizado

Segundo Gonzalez (2000), este método aplica à imagem final parte dos canais verde e azul da imagem esquerda, mantendo inalterados os componentes azul e verde da imagem direita:

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 0 & 0{,}7 & 0{,}3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix}$$

Este método permite obter:

Reprodução parcial de cores, porém perdendo os tons de vermelho do original;

Rivalidade retínica bastante reduzida.

Figura 29. Anaglifo otimizado.

4.10. Paralaxe

A paralaxe é a distância medida em relação a um anteparo de referência (que pode ser a tela do computador, por exemplo). As medidas de paralaxe podem ser classificadas em três tipos: zero, positiva e negativa (RAPOSO, et al. 2004).

Paralaxe negativa: os eixos dos olhos se cruzam antes do anteparo Figura 30.

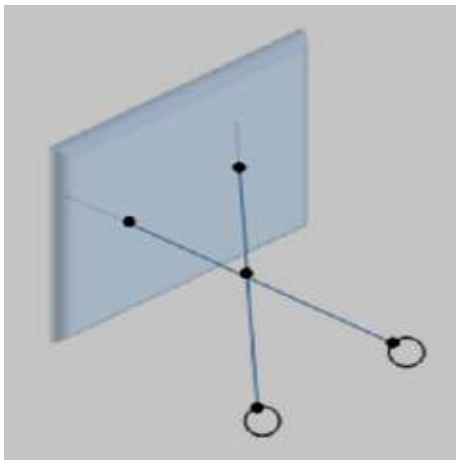

Figura 30. Paralaxe negativa.
Fonte: Gonzalez (2000).

Paralaxe zero: os eixos dos olhos se cruzam no anteparo Figura 31.

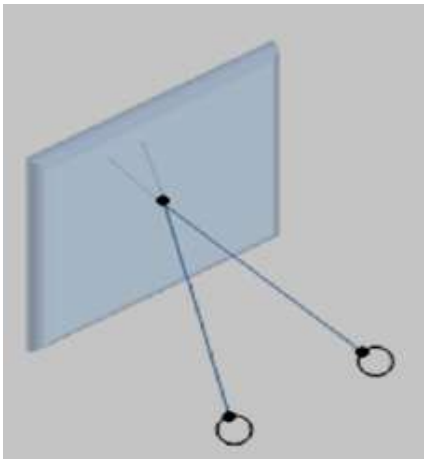

Figura 31. Paralaxe zero.
Fonte: Gonzalez (2000).

Paralaxe positiva: os eixos dos olhos se cruzam após o anteparo Figura 32.

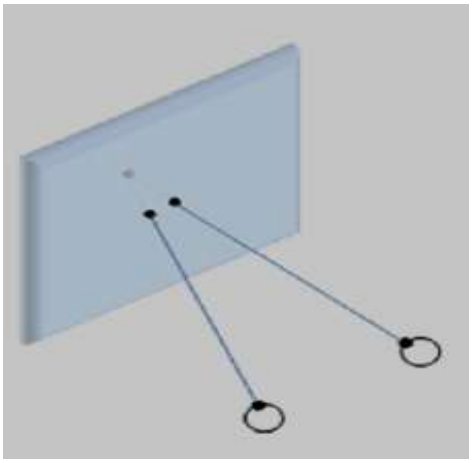

Figura 32. Paralaxe positiva.
Fonte: Gonzalez (2000).

A paralaxe faz com que as imagens vistas por cada olho sejam diferentes provocando a disparidade da retina, que por sua vez provoca o fenômeno da estereoscopia (RAPOSO, et al. 2004).

Pode ser definida em termos de medida angular, relacionando a com a disparidade e com a distância do observador a tela de exibição. Se as projeções de um ponto no olho esquerdo e direito estão separadas por P centímetros e o observador está a D centímetros da tela, então o ângulo de paralaxe α é dado por (ALBUQUERQUE, 2006):

$$\alpha = \arctan\frac{P}{D}.$$

Em geral o valor do ângulo de paralaxe deve estar no intervalo [-1,5°, 1,5°] Figura 33, definindo valores de paralaxes mínimos e máximos. Valores fora deste intervalo tendem a causar desconforto na visualização (ALBUQUERQUE, 2006).

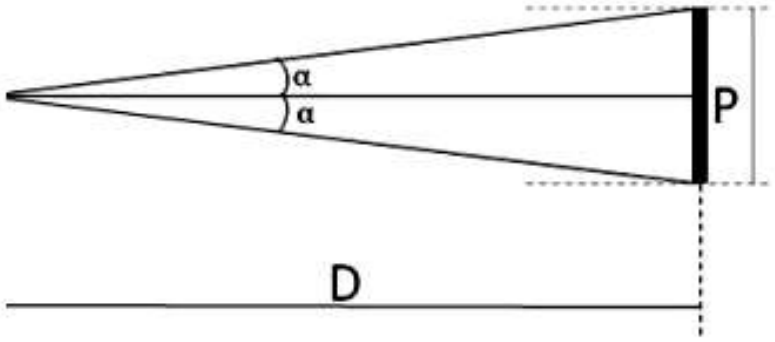

Figura 33. Valores de paralaxe.
Fonte: Albuquerque (2006).

Se a paralaxe positiva tem um valor menor, mas próximo da distância entre os olhos, o seu resultado é ruim, a menos que se queira posicionar o objeto no infinito. Se P for maior que a distância entre os olhos, há um erro, pois neste caso os olhos estariam que estar divergindo (RAPOSO, 2004).

A estereoscopia não ocorre se apenas um olho enxergar o ponto. Para evitar que isso aconteça, deve atentar-se para que as projeções ajustem-se no retângulo que define o campo de visão no plano de projeção (RAPOSO, 2004).

4.11. Armazenamento de imagens

Segundo Date (2000), o armazenamento de imagens digitais é um dos desafios no projeto de sistemas de processamento de imagens, em razão da grande quantidade de *bytes* necessários para este processo. O armazenamento pode ser dividido em três categorias: armazenamento de curta duração de uma imagem, enquanto ela é utilizada nas outras etapas do processamento, armazenamento de massa para operações de recuperação de imagens relativamente rápidas e arquivamento de imagens, para recuperação futura quando isto se fizer necessário. O espaço de armazenamento requerido é normalmente especificado em *bytes* (8 *bits*). Para o armazenamento de curta duração, a

alternativa mais simples é utilizar parte da memória *RAM*. Outra opção consiste no uso de placas especializadas, chamadas *frame buffers*, que armazenam uma ou mais imagens completas e podem ser acessadas a uma alta velocidade, tipicamente 30 imagens completas por segundo. O uso de *frame buffers* permite também que operações de *zoom* (ampliação ou redução para fins de visualização), *scroll* (rolagem na vertical) e *pan* (rolagem na horizontal) sejam executadas de forma praticamente instantânea. A segunda categoria de armazenamento normalmente discos magneto-óptico. Nesta categoria o fator tempo de acesso é tão ou mais importante que a capacidade (em *bytes*) do meio de armazenamento. O arquivamento de imagens é caracterizado por altas quantidades de *bytes* contendo imagens cuja recuperação é esporádica. Nesta categoria temos os discos ópticos WORM (*Write-Once-Read-Many*), com capacidade que pode chegar a mais de 10 GB por disco (DATE, 2000).

As bases de dados de tamanhos diversos gerenciam nossas vidas, como exemplos simples a conta bancária de uma pessoa a qual faz parte de uma coleção de contas bancárias do banco que se trabalha. Também podemos citar como exemplo o título eleitoral ou cadastro de pessoa física de determinada pessoa, os quais certamente estão armazenados em bancos de dados de grande porte. Quando sacamos dinheiro no caixa eletrônico de nosso banco, nosso saldo e as movimentações existentes em nossa conta bancária já estão à nossa disposição. Nestas situações sabemos que existe uma necessidade em se realizar o armazenamento de uma série de informações que não se encontram efetivamente isoladas umas das outras, ou seja, existe uma ampla quantidade de dados que se referem a relacionamentos existentes entre as informações a serem manipuladas (DATE, 2000).

Segundo Date (2000), estes bancos de dados, além de manterem todo este volume de dados organizado, também devem permitir atualizações, inclusões e exclusões do volume de dados, sem nunca perder a consistência. Um sistema de banco de dados é basicamente um sistema computadorizado de armazenamento de registros, isto é, um sistema computadorizado cujo propósito geral e armazenar informações e permitir ao usuário buscar e atualizar essas informações quando solicitado. As informações em questão podem ter qualquer significado para o individuo ou a organização a que o sistema deve servir, tudo o que seja necessário para auxiliar no processo geral da tomada de decisões de negócios desse indivíduo ou dessa organização.

4.12. Processamento de imagens

Segundo Marques (1999), o processamento de imagens digitais envolve procedimentos normalmente expressos sob forma algorítmica. Em função disto, com exceção das etapas de aquisição e exibição, a maioria das funções de processamento de imagens pode ser implementada via *software*. O uso de *hardware* especializado para processamento de imagens somente será necessário em situações nas quais as limitações do computador principal (por exemplo, velocidade de transferência dos dados através do barramento) forem intoleráveis.

A tendência atual do mercado de hardware para processamento de imagens é a comercialização de placas genéricas compatíveis com os padrões de barramento consagrados pelas arquiteturas mais populares de microcomputadores e estações de trabalho. O *software* de controle destas placas é que determinará sua aplicação específica a cada situação. As vantagens mais imediatas são: redução de custo, modularidade, reutilização de componentes de software em outra aplicação rodando sobre o mesmo *hardware* e independência de fornecedor. Convém notar, entretanto, que sistemas dedicados continuam sendo produzidos e comercializados para atender a tarefas específicas, tais como processamento de imagens transmitidas por satélites (MARQUES, 1999).

As informações geográficas existentes nos mapas são trabalhadas através de técnicas matemáticas e computacionais pelo geoprocessamento. As primeiras tentativas de automatizar o processamento de dados com características espaciais aconteceram nos anos de 1950, mas, foi na década de 80, com os avanços da microinformática e estabelecimento de centros na área computacional que o geoprocessamento começou sua maior projeção (CÂMARA et al., 2002).

Segundo Teixeira (1992), o software usado para o geoprocessamento é normalmente o SIG (Sistema de Informações Geográficas) o qual permite a integração de diferentes mapas temáticos e uma gama de cálculos. Esses sistemas se constituem de uma série de programas e processos de análise, cuja característica principal é ajustar o relacionamento de determinado fenômeno da realidade com sua localização espacial.

O SIG fundamenta-se na coleta, armazenamento, recuperação, análise e tratamento de dados espaciais, não espaciais e temporais, auxiliando as tomadas de decisões e dando suporte às diferentes atividades como, por exemplo, no

gerenciamento, análise e planejamento de bacias hidrográficas e aplicação em diversas áreas de conhecimento, podendo ser utilizado desde uma simples divisão territorial até grandes projetos de gerenciamento de banco de dados (CRUZ, 2003).

O SIG foi principalmente projetado para a manipulação de dados espaciais, portanto, todo e qualquer dado considerado como espacial, como as redes de drenagem pode ser mapeável, isso é, toda informação espacial deve estar ligada a um objeto específico em um mapa e, a localização nesse mapa, deve ser referenciada geograficamente. A apresentação dos resultados de um SIG pode ser feita através da produção de textos, tabelas ou mapas, contendo dados originais ou processados, possibilitando a análise espacial de um fenômeno. As características inerentes a esses sistemas, especialmente pela capacidade de tratar de forma integrada e manusear grandes quantidades de dados, o SIG é uma ferramenta importante na elaboração de estudos (TEIXEIRA, 1992).

4.13. Exibição de imagens

Segundo Teixeira (1992), o monitor de vídeo é um elemento fundamental de um sistema de processamento de imagens. Os monitores em uso atualmente são capazes de exibir imagens com resolução de pelo menos 640 x 480 *pixels* com 256 cores distintas, 800 x 600 *pixels* com 512 cores distintas e 1024 x 768 *pixels* com 1024 cores distintas.

4.14. Sistemas de gerenciamento de informações e apoio a decisão

Segundo Sprague & Hugh (1991), (SAD) sistemas de apoio à decisão é qualquer sistema de gerenciamento de informação, que forneça informações, para auxílio à tomada de decisão. Sistemas de gerenciamento de informações, através do processo de gerenciamento, podem contribuir para o processo qualidade da informação e ao processo de apoio a tomada de decisão minimizando os erros ou dúvidas.

SAD, que também são conhecidos como *"Decision Support Systems"* (DSS), possuem funções específicas, não vinculadas aos sistemas existentes, que permitem buscar informações nas bases de dados existentes e delas retirar subsídios para o processo

de decisão. SAD começam a ser desenvolvido nas organizações a partir dos estágios de controle e integração no modelo proposto segundo (NOLAN, 1977).

4.15. Desenvolvimento de sistemas computacionais para a área agrícola

A utilização das novas tecnologias de informação e comunicação na agricultura, aliadas a gestão, administração e economia têm dado origem a novos e interessantes programas de computador. A quantidade de programas de computador desenvolvidos para o setor de agronegócios aumenta gradativamente a cada ano (BARBOSA ET AL., 2000).

A tecnologia da informação começou a ser aplicada com sucesso nas fazendas com a automatização das tarefas de contabilidade, de controle de recursos humanos e de controle de estoques e de maquinário (MEIRA et al., 1996).

4.16. Realidade aumentada

Ao contrário da computação gráfica, a relaidade aumentada é um campo de pesquisa ainda recente e pouco desenvolvido. O termo realidade aumentda foi utilizado pela primeira vez apenas em meados de 1990, apesar de alguns dispositivos de visualização estereoscópica e interação com objetos virtuais já existirem na ocasião. Na publicação original onde o termo foi pela primeira vez definido (WELLNER; MACKAY; GOLD, 1993), o mesmo foi introduzido como sendo o oposto da Realidade Virtual. Enquanto a Realidade Virtual imerge o usuário em um mundo sintético, formado puramente de informação, a meta da RA é amplificar o mundo real com a capacidade de processar informações (WELLNER; MACKAY; GOLD, 1993).

Desde os anos 90, o termo realidade aumentada tem aparecido com freqüência na literatura científica. Entretanto, o seu uso se deu em diferentes contextos e com propósitos diversos, fazendo com que várias definições para o termo fossem formuladas, sem que nenhuma fosse especialmente mais aceita ou mais adequada (MILGRAM, et al. 1994).

Por exemplo, uma definição bem abrangente de realidade aumentada seria “o aumento do *feedback* natural do usuário utilizando sinais simulados”. Uma outra definição desnecessariamente restritiva seria a de “uma forma de realidade virtual onde o

headmounted display do usuário é transparente, permitindo uma visão clara do mundo real" (MILGRAM, et al. 1994).

Considera-se a realidade aumentada como parte de um universo mais amplo, a realidade misturada. Espectro de dois extremos, o ambiente puramente real e o ambiente puramente virtual. Neste contexto, a realidade misturada consiste de um ambiente localizado entre estes dois extremos, onde objetos reais são apresentados juntamente com objetos virtuais em um mesmo dispositivo de exibição ou *display*. Dentro desta classificação as aplicações de realidade misturada podem ser caracterizadas em duas classes principais: a Virtualidade Aumentada, onde os elementos virtuais predominam sobre o mundo real e a RA, onde os elementos reais são predominantes sobre os elementos virtuais (KIRNER e TORI 2004) e (MILGRAM, et al. 1994).

A Figura 34 mostra o espectro de Realidade Virtual (*Virtual Reality Continuum*).

Figura 34. Espectro de realidade virtual.
Fonte: Adaptado de (MILGRAM, et al. 1994)

Entre os extremos deste espectro encontramos vários tipos de sistema; o sistema que vai nos interessar neste projeto é um sistema de realidade aumentada baseado em uma definição adaptada de R. T. Azuma (1997). Este sistema de realidade aumentada possui as seguintes características:

1) Combinação do ambiente real com elementos virtuais;

2) Interação do usuário com os elementos virtuais em tempo real;

3) Os elementos virtuais e o ambiente real devem estar alinhados espacialmente (isto é, possuir um sistema de coordenadas espaciais em comum) em três dimensões.

Esta definição retém os componentes essenciais da realidade aumentada, sendo abrangente o suficiente para englobar uma variedade de tecnologias e restritivo o suficiente para eliminar aplicações que não seriam de interesse do campo da realidade aumentada. Filmes como "Jurassic Park", por exemplo, incluem em suas cenas

objetos virtuais realistas misturados a um ambiente real em 3D, porém não constituem uma mídia interativa. A sobreposição bidimensional de objetos virtuais pode ser feita sobre vídeo em tempo real e com interatividade, porém os objetos virtuais estão alinhados com o mundo real em apenas duas dimensões, não constituindo uma aplicação de realidade aumentada. Entretanto, esta definição inclui interfaces baseadas em monitores, sistemas monoculares, HMDs, e várias outras técnicas de mistura de cenas (R. T. AZUMA 1997).

A Figura 35 apresenta a combinação do ambiente real com elementos virtuais de acordo com a definição apresentada.

Figura 35. Realidade aumentda com vaso e carro virtuais sobre a mesa.
Fonte: (Tori, Kirner e Siscoutto, 2006).

Os sistemas de realidade aumentada apresentam requisitos mais restritos em comparação aos requisitos impostos pelas tecnologias de realidade virtual ou virtualidade aumentada desenvolvidas, como o posicionamento preciso dos elementos virtuais sobre a cena real (registro), boa percepção de profundidade, procedimentos simples de configuração inicial, baixa restrição da liberdade de movimentos do usuário, e baixa latência no cálculo e exibição de imagens (STATE, et al. 1997).

4.16.1. Visão direta x visão indireta

Segundo Tori, Kirner e Siscoutto (2006), os ambientes com realidade aumentada podem ser classificados em dois tipos principais, de acordo com a forma de visualização da cena misturada. Nos casos em que o mundo misturado é visualizado diretamente através dos olhos do usuário, a realidade aumentada é classificada como imersiva ou de visão direta. Nos casos em que a mistura das cenas real e virtual ocorre em um dispositivo como um monitor ou projetor estático, a RA é classificada como não

imersiva ou de visão indireta. Podemos concluir que a principal diferença entre os sistemas de visão direta e os de visão indireta, diz respeito à interatividade da superfície de projeção com relação ao observador. Nos sistemas de visão direta, a superfície de projeção é interativa ao observador, enquanto nos sistemas de visão indireta a superfície de projeção é estática.

4.16.2. O problema do registro

Um objetivo básico de qualquer sistema de realidade aumentada é o alinhamento preciso dos objetos virtuais sobre os objetos reais, utilizando um sistema de coordenadas tridimensionais comum aos dois universos. Isto exige, entre outras coisas, sensores que sejam precisos em tempo real para calcular a posição e orientação do usuário e dos objetos de interesse. Em um cenário ideal, um sistema de realidade aumentada deve ser capaz de efetuar registro com precisão milimétrica em ambientes externos e sem preparação prévia do ambiente. (HOFF e AZUMA, 2000).

Existem duas abordagens básicas que podem ser combinadas ou utilizadas separadamente para solucionar o problema do registro em aplicações de realidade aumentada: através de sensores ou por técnicas de visão computacional, sendo que esta última vem ganhando bastante força com o aumento na disponibilidade de poder computacional. A visão computacional consiste de técnicas de reconhecimento de imagens por computador, que permitem obter informações sobre a posição de objetos cuja aparência possa ser modelada computacionalmente; desta forma, o vídeo captado pelas câmeras do sistema de realidade aumentada é analisado em busca dos padrões de imagem que possibilitam o cálculo da posição dos objetos da cena. (HOFF; AZUMA, 2000)

4.16.3. Realidade aumentada espacial

Dentro do campo de realidade aumentada, a realidade aumenteada espacial abrange as técnicas de realidade aumentada que realizam a mistura dos elementos reais e virtuais no próprio espaço físico do usuário, sem a utilização de dispositivos de visualização especiais como os *Head Mounted Displays*. Os elementos virtuais podem ser projetados sobre uma superfície física (RASKAR, 1998).

A detecção do ponto de vista do usuário permite que as imagens possam ser atualizadas dinamicamente criando a ilusão de que os elementos virtuais estão alinhados ao mundo real mesmo com o observador em movimento. Um dos grandes problemas encontrados por aplicações de realidade aumentada espacial é a sua grande dependência das características da superfície de projeção. Uma superfície que possui baixa reflexão, por exemplo, dificulta a visualização das imagens. Existe também a interferência das sombras do próprio observador, que podem atrapalhar a projeção, agravando se ao permitir que haja vários usuários no mesmo ambiente (RASKAR, 1998).

5 MATERIAL E MÉTODOS

Para contextualizar o estudo e aprofundar o conhecimento sobre o tema, foi utilizada a pesquisa e revisão bibliográfica, bem como pesquisa de trabalhos cientificos, artigos e livros para a elaboração da redação da dissertação. Foi realizado o levantamento dos requisitos, analise, desenvolvimento e implementação de um sistema computadorizado para manipulação de imagens obtidas por fotografia aérea. Foi confrontado programas computacionais utilizados com o programa computacional desenvolvido.

Cervo & Bervian (1996) descrevem a pesquisa bibliográfica como aquela que procura explicar um problema a partir de referências teóricas publicadas em documentos.

Segundo Gil (2002), a pesquisa bibliográfica é desenvolvida com base em material já elaborado, constituído principalmente de livros, teses e artigos científicos. Segue o autor, que este tipo de pesquisa mostra-se muito prática por permitir ao pesquisador cobrir uma gama de fatos muito mais ampla do que através de investigação direta.

A elaboração, desenvolvimento e implementação contenplou as seguintes etapas:

1ª etapa: Revisão bibliográfica, levantamento dos conceitos e técnicas para a elaboração da tese e do sistema.

2ª etapa: Levantamento dos requisitos, Analise do sistema, Desenvolvimento e Implementação.

3ª etapa: Confrontar os programas computacionais atualmente utilizados com o programa computacional desenvolvido de modo a identificar os pontos positivos e negativos de sua utilização.

5.1. Descrição da área de estudo

O aplicativo foi desenvolvido no laboratório de computação dos alunos de pós-graduação do Departamento de Engenharia Rural na Faculdade de Ciências Agronômicas - FCA, da Universidade Estadual Paulista "Júlio de Mesquita Filho" - UNESP, Campus de Botucatu, Estado de São Paulo.

O levantamento das necessidades e requisitos do sistema foi executado na forma de pesquisa de campo usando como referência as necessidades das disciplinas que envolvem fotointerpretação, fotogrametria, fotografias aéreas no mapeamento da oupação do solo, avaliação e uso de bacias hidrográficas, além da revisão da literatura e avaliação de sistemas e técnicas já existentes.

5.2. Materiais

O microcomputador responsável pelo armazenamento e processamentos dos dados foi um microcomputador Intel® Core 2 Duo 2.4GHz, 4Gb de RAM, HD 1TB 7200 RPM, sistema operacional Windows XP Professional 2002, com acesso direto a internet.

O microcomputador responsável pelo desenvolvimento e teste do aplicativo foi um microcomputador notebook AMD® Athlon 64, 1.6 GHz, 2Gb de RAM, HD 160GB, sistema operacional Windows 7 Professional.

5.3. Linguagem computacional

As rotinas foram desenvolvidas de forma convencional, porém priorizando sempre a acessibilidade por usuários não especializados na área da informática,

com acesso a diversos módulos do sistema através de *menus*, tornando simplificado e de fácil operação o sistema.

Os programas computacionais escolhidos para o desenvolvimento do projeto serão Eclipse e Netbeans para a etapa de programação do aplicativo, com acesso ao banco de dados Mysql.

A linguagem de programação JAVA foi escolhida por ser uma linguagem orientada a objetos derivada da linguagem de programação C, sendo também uma linguagem de programação gratuita, livre, levando em consideração que a linguagem possibilita o desenvolvedor utilizar compartilhamento de código, além de permitir o desenvolvimento rápido e visual de aplicativos para plataforma Windows e Linux dentre outros sistemas operacionais.

5.3.1. Linguagem Java

Segundo Gonzalez (2000), Java é uma linguagem de programação orientada a objetos, independente de plataforma, que foi desenvolvida pela *Sun Microsystems, Inc*. Atualmente, é uma das linguagens mais utilizadas para o desenvolvimento de sistemas, e pode ser obtida gratuitamente em http://java.sun.com. Java é tanto compilada como interpretada: o compilador transforma o programa em *bytecodes*, que consiste em um tipo de código de máquina específico da linguagem Java; o interpretador, disponível na JVM (*Java Virtual Machine*) que pode ser instalada em qualquer plataforma, transforma os *bytecodes* em linguagem de máquina para execução, sem que seja necessário compilar o programa novamente.

O ambiente Java engloba tanto um compilador, quanto um interpretador. Somente para executar programas Java é utilizado o JRE (*Java Runtime Environment*), que normalmente é instalado junto com as versões mais recentes dos navegadores para Internet, para possibilitar a execução de *applets*. Entretanto, para o desenvolvimento de novos programas, é preciso instalar o J2SE (*Java 2 Platform, Standard Edition*) que inclui o compilador, a JVM e a API (*Application Programming Interface*). A API básica que é instalada com o J2SE, engloba os pacotes que contêm as classes responsáveis pelas funcionalidades de entrada e saída, interface gráfica, coleções, entre outras. Além disso, existe a *Java Standard Extension API*, que inclui outros pacotes,

tais como acesso a banco de dados e o *Java Media Framework*, que suporta tecnologias gráficas e multimídia. Java 3D é um dos componentes deste pacote (SANTOS, 2004).

5.3.2. Conceitos básicos de computação gráfica com Java

Para entender a API Java 3D e aproveitar melhor as suas funcionalidades, é importante conhecer alguns conceitos básicos de computação gráfica. Quando se trabalha com síntese de imagens 3D, deve-se considerar que a maioria dos periféricos de entrada e saída é 2D. Portanto, várias técnicas são utilizadas para contornar estas limitações dos dispositivos e possibilitar a visualização com realismo de um objeto modelado, por exemplo, através de uma malha de triângulos. Transformações geométricas de escala, rotação e translação, manipulação de uma câmera sintética, projeção perspectiva, iluminação, cor, sombra e textura são algumas das técnicas utilizadas para contribuir na geração de imagens de alta qualidade (SANTOS, 2004).

As transformações geométricas consistem em operações matemáticas que permitem alterar uniformemente o aspecto de objetos, mas não a sua topologia. São usadas para posicionar, rotacionar e alterar o tamanho dos objetos no universo. O conceito de câmera sintética é usado para definir de que posição o objeto será visualizado, como se fosse obtida uma "foto" quando a câmara estava numa dada posição direcionada para o objeto. Neste processo, torna-se necessário aplicar uma projeção, que é o procedimento utilizado para se obter representações 2D de objetos 3D.Várias técnicas também foram desenvolvidas para tentar reproduzir a realidade em termos de aparência, por exemplo, efeitos de iluminação e de textura de materiais. Portanto, diversas fontes de luz, tais como pontual e spot, podem ser incluídas em um universo para permitir a simulação da reflexão dos objetos, que possibilita descrever a interação da luz com uma superfície, em termos das propriedades da superfície e da natureza da luz incidente (GONZALEZ, 2000).

Além disso, cada tipo de material tem características próprias, que permitem sua identificação visual ou táctil. Por exemplo, microestruturas podem produzir rugosidade na superfície dos objetos, ou estes podem ser compostos por um material como mármore ou madeira. As técnicas usadas para simular estes efeitos em Computação Gráfica são conhecidas como mapeamento de textura. Todas estas técnicas foram implementadas na API Java 3D (SANTOS, 2004).

5.3.3. Instalação e utilização

Segundo Gonzalez (2000), todo software necessário para trabalhar com Java 3D é gratuito e pode ser obtido no site da Sun (http://java.sun.com). A primeira etapa consiste em instalar o J2SE, versão 1.2 ou posterior .Num segundo momento, deve-se instalar a API Java 3D, que está disponível em http://java.sun.com/products/java-media/3D/download.html. Como Java 3D é baseada nas bibliotecas OpenGL ou DirectX (figura 36), estas também devem estar instaladas.

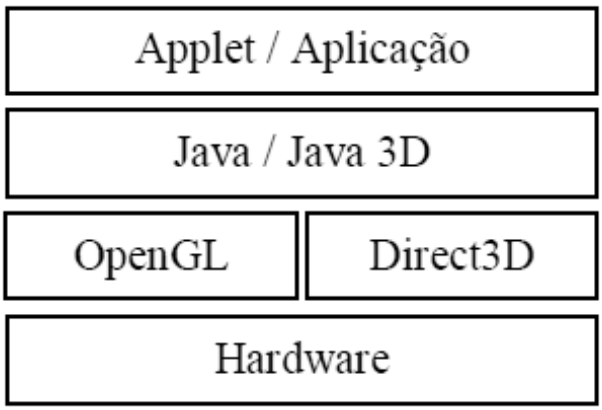

Figura 36. Camadas de software existentes quando se utiliza Java 3D.
Fonte: Gonzalez (2000).

Segundo Gonzalez (2000), o processo de compilar e de executar é o mesmo que para aplicações e *applets* Java, isto é, utiliza-se o comando *javac FileName.java* para compilar, e *java FileName* para executar através do interpretador. Entretanto, outros pacotes, referentes a API Java 3D, devem ser importados no código-fonte das classes (por exemplo, *com.sun.j3d.utils.** e *javax.media.j3d.**). Para facilitar o desenvolvimento, recomenda-se a utilização de um ambiente de programação.

5.3.4. APIs Java para imagens

A comunidade de processamento de imagens era formada por um grupo pequeno de pesquisadores que tinham acesso limitado a ferramentas comerciais de PDI, algumas desnecessárias, e que eram desenvolvidas sob componentes de softwares proprietários. Estes eram resumidos a pequenos pacotes de *software* para carregamento de arquivos de imagem e armazenamento dos mesmos em disco e ainda apresentavam

documentações incompletas e formatos de arquivos proprietários (MURRAY; VANRYPER, 1996). Surgiram então novos formatos, inicialmente desenvolvidos para aplicações específicas, mas muitos dos quais não despontaram ou foram esquecidos (MIANO, 1999). Nos anos 80 e 90 surgiram alguns novos formatos de imagens e diversos *softwares* para convertêlos, além do suporte dos formatos criados para modelos específico de *hardware* definidos pelos desenvolvedores.

Um pequeno conjunto desses formatos permaneceu na ativa: *Tagged Image File Format* (TIFF), *Graphics Interchange Format* (GIF), *Portable Network Graphics* (PNG), padrão JPEG, *Windows Bitmap* (BMP) e o *Portable Bitmap Format* (PBM) que possui os tipos PBM (*portable bitmap*) para *bitmaps* binários, PGM (*portable graymap*) para imagens em nível de cinza e PMN (*portable any map*) para imagens coloridas (BURGER & BURGE, 2009). Esses são suportados pela maioria das APIs padrões para C/C++ e Java.

Com o incremento da capacidade de compressão dos formatos de arquivos de imagem e a evolução das linguagens de programação, muitos desenvolvedores têm definido seus aplicativos utilizando a linguagem de programação Java para sistemas de PDI (RODRIGUES, 2001). A linguagem possui algumas características que têm sido atrativas à comunidade de desenvolvimento (DEITEL; DEITEL, 2005) como: é livre, portável ("*write once, run anywhere*"), segura, sintaxe similar à linguagem C, possui facilidade de internacionalização dos caracteres, vasta documentação, coletor de lixo (para desalocação automática de memória) e facilidade de criação de programação distribuída e concorrente. Há ainda o paradigma orientado a objetos (SANTOS, 2003) que define o modus operandi de Java e descreve a estrutura de todas as APIs criadas a partir desta linguagem.

Para representação e processamento de imagens, a Java *Advanced Imaging* (JAI) API (criada e mantida pela Sun Microsystems) pode ser trabalhada. Ainda que a API não faça parte de um software de processamento digital de imagens completo, as operações existentes e possibilidades de extensão aliadas ao baixo custo e implementação simples tornam esta API uma opção atrativa para desenvolvimento de algoritmos de processamento de imagens (SANTOS, 2004).

Nesta API, a *PlanarImage* é a principal classe para representação de imagens na JAI e traz mais flexibilidade que a classe nativa *BufferedImage*. Em ambas as classes, seus *pixels* são armazenados em uma instância de Raster que contém uma instância

de uma subclasse de *DataBuffer* (*DataBufferByte, DataBufferFloat, DataBufferInt, DataBuffer- Double, DataBufferShort*, etc.), de acordo com o comportamento seguido pela instância de uma subclasse de *SampleModel*. Uma instância de *PlanarImage* também tem uma instância de *ColorModel* associada a ela, a qual contém uma instância de *ColorSpace*, que determina como os valores de *pixels* serão deslocados aos valores de cores (SANTOS, 2004).

Figura 37 mostra a composição geral da classe *PlanarImage* com as demais classes da API Java 2D.

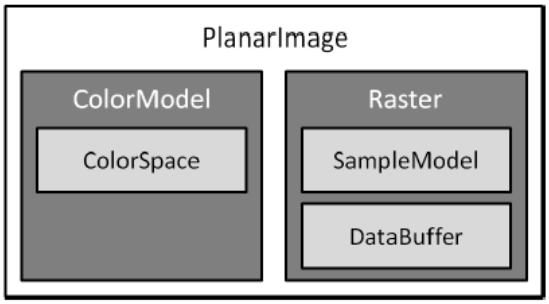

Figura 37. Camadas Estrutura da classe de suporte a imagem na API JAI.
Fonte: Santos (2004).

Uma *PlanarImage* permite somente leitura à imagem, ou seja, pode ser criada e seus valores de *pixels* podem ser lidos de várias maneiras, mas não há métodos implementados para a modificação de seus elementos (SANTOS, 2004). *PlanarImage* possui uma grande flexibilidade já que pode ter a origem da imagem em uma posição diferente da coordenada (0,0), ou ainda coordenadas de *pixels* com valores negativos (RODRIGUES, 2001).

Figura 38. Imagem.
Fonte: Rodrigues (2001).

Para visualização, a API JAI disponibiliza um componente simples e extensível determinado pela classe *DisplayJAI*. Esta é uma subclasse de *JPanel* e pode ser trabalhado como qualquer outro componente gráfico Java. Na Tabela 1, é apresentado o código fonte referente a uma classe de exibição de imagem chamada ExibeImagem que faz uso de uma instância de *DisplayJAI.* Essa mesma instância é associada com um objeto de *JScrollPane* para suportar (com barras de rolagem) qualquer imagem que possua dimensão maior que a interface gráfica definida. O resultado da execução da classe ExibeImagem está disposto na Figura 38.

```
/**
 * Lendo imagem usando apenas APIs nativas
 *
 * @author GALVAO
 */
public class ExibeImagem extends JFrame {

   // Construtor padrão: define interface e exibe a imagem lida na mesma

   public ExibeImagem() throws IOException {
      BufferedImage imagem = ImageIO.read(new File("Lenna.png
      "));
         String infoImagem = "Dimensões: "+imagem.getWidth() +
         "x"+imagem.getHeight() +
      " Bandas: "+
         imagem.getRaster().getNumBands();
      ImageIcon icone = new ImageIcon(imagem);
      JLabel labImagem = new JLabel(icone);
      this.setTitle(
      "Display da Imagem: "+"Lenna.png
      ");
         Container contentPane = this.getContentPane();
      contentPane.setLayout(new BorderLayout());
      contentPane.add(new JScrollPane(labImagem), BorderLayout.CENTER);
      contentPane.add(new JLabel(infoImagem), BorderLayout.SOUTH);
      this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      this.setSize(imagem.getWidth(), imagem.getHeight());
      this.setVisible(true);
   } // fim do construtor
   // Método principal

   public static void main(String args[]) {
      try { // tratamento de exceção de E/S
         ExibeImagem appExibeImg = new ExibeImagem();
      } catch (IOException exc) {
         System.out.println(
```

```
        "Erro de leitura
        ! "+exc.getMessage()
);
        }
        } // fim do método principal
        } // fim da classe
```

Tabela 1 - Código da classe ExibeImagem.

Uma possibilidade para o processamento de imagens com Java é o uso da API e software *ImageJ* (RASBAND, 2009). Criada por Wayne Rasband, ela apresenta em seu site oficial (http://rsbweb.nih.gov/ij/) a versão atual do aplicativo, pacotes, documentação, updates, código fonte completo, imagens para teste e uma coleção continuamente crescente de plugins (BURGER & BURGE, 2009) desenvolvidos por terceiros e que podem ser adicionados ao aplicativo. A organização da API *ImageJ* está disposta na Figura 39. Algumas classes da API são definidas a partir da API AWT de interface gráfica. Isso permite que os resultados de suas operações sejam visualizados também pela API *Swing* da linguagem Java, porque essa última herda as características de AWT. Com relação aos plugins, os desenvolvedores podem criar seus próprios ou reaproveitar plugins já utilizados no aplicativo. Isso dá uma grande flexibilidade na criação de classes mas não impede a produção de códigos redundantes entre os trabalhos de vários desenvolvedores.

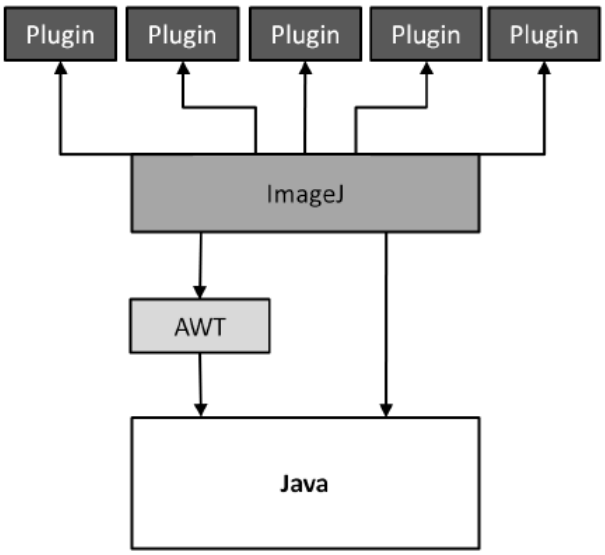

Figura 39. Estrutura geral da API ImageJ.
Fonte: Burger & Burge (2009).

As classes nativas da API oferecem suporte à maioria das técnicas de PDI. Duas classes são bastante utilizadas: *ImagePlus* e *ByteProcessor*. A classe *ImagePlus* normalmente exige um objeto de *java.awt.Image* na instanciação de seus objetos e sempre carrega a imagem trabalhada para a visualização na interface gráfica (associando com a classe ImageCanvas). Com relação à classe *ByteProcessor*, essa é a base da execução das principais técnicas de processamento de imagens com a *ImageJ*. A partir de uma instância dessa classe é possível chamar diretamente um método para muitas das operações a serem vistas na próxima seção. O aplicativo *ImageJ* foi desenvolvido para teste das operações implementadas pela API e pelos plugins mas ele tem sido utilizado também em uma abordagem didática para exibição dos resultados. Esta seção apresentou uma visão geral sobre as API JAI e *ImageJ*. Essas são as principais referências para processamento de imagens usando a linguagem de programação Java. Suas definições e arquitetura foram dispostas aqui, mas o modo de uso de seus operadores (BURGER & BURGE, 2009).

5.3.5. O pipeline gráfico do OpenGL

O OpenGL (*Open Graphics Library*) é uma interface de software para manipulação do hardware gráfico de um sistema. Ele expõe funcionalidades avançadas de renderização do hardware gráfico mantendo um modelo de programação simples (SEGAL; AKELEY, 2006).

Sob a ótica do programador, o OpenGL é um conjunto de comandos que permite a especificação de objetos em duas ou três dimensões, juntamente com comandos que controlam a renderização destes objetos no framebuffer (SEGAL; AKELEY, 2006).

Sob a ótica do implementador (fabricante do *hardware* gráfico), o OpenGL é um conjunto de comandos que afeta a operação do *hardware* gráfico. Se o *hardware* consistir de apenas um *framebuffer* manipulável, o OpenGL deve ser implementado em sua maior parte na unidade central de processamento do sistema. Tipicamente, o hardware gráfico pode incluir componentes de aceleração gráfica em graus variados, desde um subsistema de rasterização capaz de renderizar linhas e polígonos bidimensionais até sofisticados processadores de ponto flutuante capazes de transformar e computar dados geométricos. A função do implementador do OpenGL é fornecer a interface de *software* especificada, dividindo o esforço necessário para cada comando do

OpenGL entre a unidade central de processamento e o *hardware* gráfico (SEGAL; AKELEY, 2006).

Sob a ótica dos seus criadores, o OpenGL é uma máquina de estados que controla um conjunto de operações de renderização. O modelo de máquina de estados deve oferecer uma especificação satisfatória para os programadores e implementadores, mas não é necessariamente um modelo de implementação. Uma implementação deve ser capaz de fornecer os resultados da maneira como especificados em (SEGAL; AKELEY, 2006), porém a forma de execução das computações não é de forma alguma ditada por esta especificação.

A funcionalidade principal oferecida pelo OpenGL consiste na transformação de primitivas geométricas (ponto, linha, polígono, *bitmap*, etc) em *pixels*, renderizando-os no *framebuffer* do *hardware* gráfico que é então exibido em tela. A figura 40 mostra o *pipeline* de processamento do OpenGL (FERNANDES, 2006).

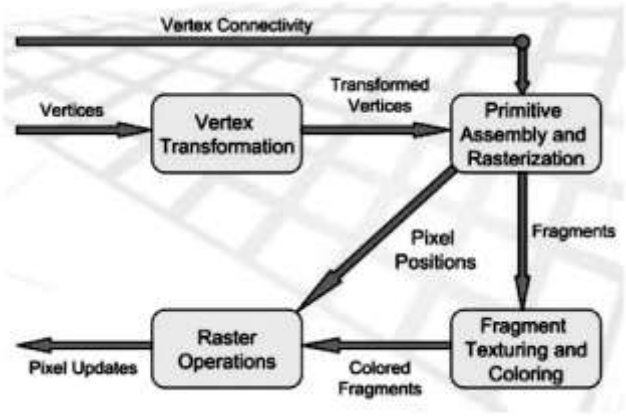

Figura 40. O Pipeline gráfico do OpenGL funcionalidade fixa.
Fonte: Burger & Burge (2009).

O *pipeline* indicado na Figura 41 mostra a funcionalidade fixa do OpenGL, que é a funcionalidade padrão oferecida pela API. Esta funcionalidade pode ser modificada com o uso de *shaders* programáveis, que substituem as operações fixas de blocos da estrutura do *pipeline*.

A sequência adequada a ser executada em cada bloco do *pipeline* é a seguinte:

a) Transformação de vértices: recebe as coordenadas e propriedades de cada vértice, bem como o estado do OpenGL, e retorna o vértice transformado

pelas matrizes de visualização (*ModelView*) e projeção (*Projection*); aplica iluminação, e gera e/ou transforma coordenadas de textura.

b) Montagem de primitivas: recebe os vértices transformados e as informações de conectividade dos vértices. Retorna as primitivas gráficas (polígonos), efetuando os cortes da cena de acordo com o volume de visualização (view frustum), e descartando as primitivas invisíveis (*back face* e Z *culling*).

c) Rasterização: Recebe as primitivas construídas no passo anterior e determina o conjunto de *pixels* cobertos por cada primitiva, bem como os atributos de cada pixel, por interpolação dos atributos entre os vértices adjacentes. Retorna um conjunto de fragmentos.

d) Texturização e coloração: Esta fase recebe os valores interpolados para cada fragmento e realiza a aplicação de texturas e coloração, calculando a cor final do fragmento.

e) Rasterização: A fase final do pipeline realiza operações como mistura (*blending*) da cor do fragmento com a cor existente em um *buffer* de cor, testes *alpha, stencil*, de profundidade, etc.

A Figura 41 fornece um resumo visual deste processo:

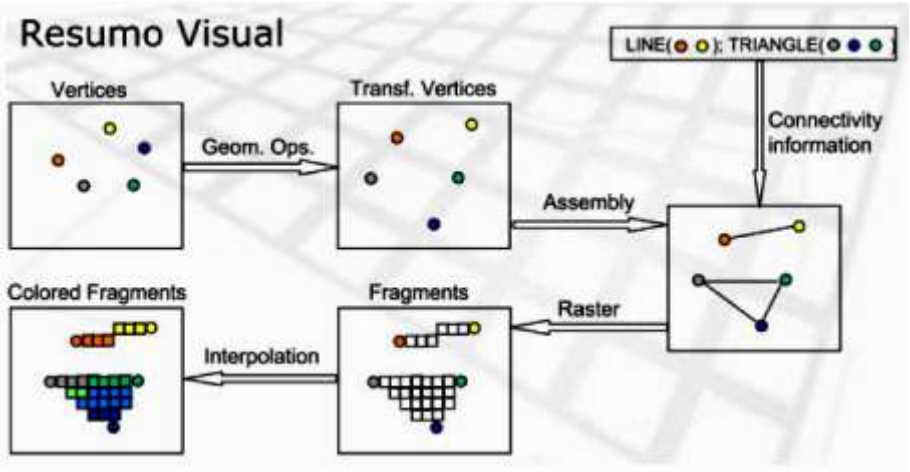

Figura 41. Resumo visual do processamento realizado pelo pipeline gráfico do OpenGL.
Fonte: Fernandes (2006).

5.3.6. Grafos de cena

O grafo de cena é a estrutura principal de um aplicativo Java3D. Um grafo de cena contém dados organizados hierarquicamente em uma árvore. O grafo de cena é composto por nós ancestrais (pais), nós terminais (filhos) e objetos de dados. Os nós ncestrais, ou group nodes, organizam e controlam como o Java 3D interpreta os seus descendentes. Os nós filho podem ser *group node*s ou *leaf nodes*, sendo que os *leaf nodes* são nós terminais e não têm filhos. Estes nós terminais determinam os aspectos semânticos principais de um grafo de cena geometria, áudio, objetos de iluminação e comportamento, e assim por diante. Os nós terminais referenciam objetos de dados do tipo *NodeComponent*, que não são nós do grafo de cena, mas contém os dados utilizados pelos nós terminais, como a geometria a ser renderizada ou as amostras de sons executadas. (Sun Microsystems, 2001)

Um grafo de cena é criado a partir de instâncias de *group nodes* e *leaf nodes*. A estrutura típica de um grafo de cena é mostrada na Figura 42. Neste diagrama podemos perceber como os subgrafos de conteúdo e visualização são separados, simplificando a manipulação da plataforma de visualização e do conteúdo adicionado ao grafo de cena.

Os objetos correspondem aos nós, ou vértices, do grafo de cena e os relacionamentos entre estes objetos são representados pelas arestas do grafo. Os relacionamentos podem ser do tipo “referência”, associando um objeto ao grafo, ou "herança”, onde um nó do tipo ‘Grupo’ pode ter um ou mais filhos e apenas um pai, e um nó do tipo ‘Folha’ não pode ter filhos (MANSSOUR, 2003).

A representação padrão de grafos de cena utiliza círculos para os nós do tipo grupo, enquanto nós do tipo folha são representados por triângulos. Os demais objetos da cena são representados por retângulos (MANSSOUR, 2003).

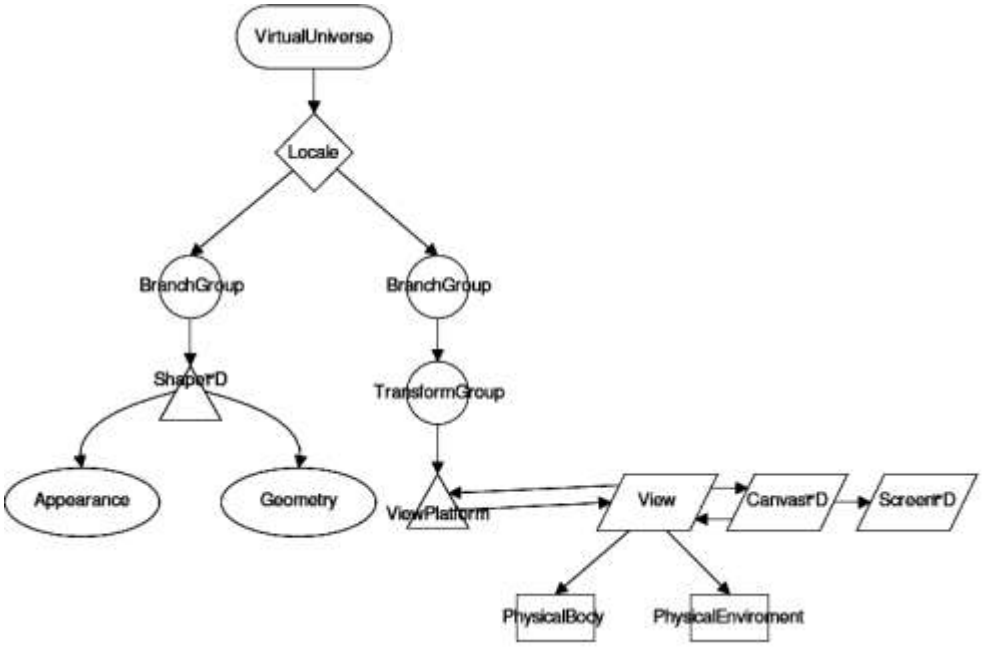

Figura 42. Grafo de cena do Java3D.
Fonte: Manssour (2003).

Um grafo de cena deve conter apenas um nó *VirtualUniverse***,** que define um universo virtual e possui pelo menos um objeto *Locale*, responsável pela especificação de um ponto de referência no universo virtual e raiz dos subgrafos de um grafo de cena (MANSSOUR, 2003).

Os nós do tipo *BranchGroup* servem para agrupar os nós de uma cena relacionados através de alguma associação ou conjunto de características em comum. (MANSSOUR, 2003).

Os sub-grafos abaixo do nó *Locale* são dois: o subgrafo de conteúdo, que descreve geometrias,aparências, comportamentos, sons e luzes, ou seja, o conteúdo do universo virtual; e o subgrafo de visualização, que controla a visualização da cena através de parâmetros como a localização do ponto de vista, a matriz de projeção, etc. Os nós do tipo *TransformGroup* especificam a posição, orientação e escala dos objetos no universo virtual, relativo a um nó Locale (MANSSOUR, 2003).

Os nós do tipo *Behavior*, não representado na figura, contém o código necessário para manipular dinamicamente a matriz de transformação da geometria de um objeto. Utilizando este tipo de objeto podemos impor transformações dinâmicas aos atributos (posição, tamanho, cor, etc.) dos objetos. Estas transformações podem ser condicionadas ao pressionamento de uma tecla, ao movimento do mouse, a uma interpolação no tempo, etc (MANSSOUR, 2003).

O nó *Shape3D* define as formas geométricas do universo virtual. Ele sempre faz referência a um ou mais objetos Geometry, que definem as formas geométricas descritas por este objeto, e a um objeto Appearance, que descreve a aparência desta geometria, com parâmetros como cor, textura, propriedades de reflexão da superfície, entre outras (MANSSOUR, 2003).

O nó *ViewPlatform* e seus objetos de dados associados descrevem o ponto de vista do observador dentro do universo virtual, e os parâmetros de renderização da cena na tela. (MANSSOUR, 2003)

5.4. Banco de dados

Foi utilizado no projeto o Mysql como servidor de banco dados para o programa computacional, sendo disponibilizado de forma gratuita e multi-plataforma, ou seja, pode se executado em diversos sistemas operacionais como Windows, Linux e Unix, considerando também sua agilidade na execução das sentenças e mínima necessidade de configuração e administração.

Além das características citadas o Mysql e capaz de atender aplicações para mono usuários, multiusuários, aplicações corporativas ou não, manipulando múltiplas bases de dados de forma confiável e independente.

5.5. Métodos

Conforme Boemo (2007), o processo de criação de um sistema informatizado compreende não só a construção de um programa de computador, mas sim varias técnicas e modelos computacionais, tais como a analise e modelagem do banco de dados que é essencial neste processo, assim como, o desenvolvimento de rotinas ou algoritmos específicos baseados nas necessidades existentes do sistema.

5.6. Processo de coleta e analise de dados antes do desenvolvimento do programa computacional

Foi obsevado as necessidades dos docentes, técnicos e discentes das áreas já citadas afim organiza-las, definindo de forma modularizada as funcionalidades do

programa computacional. Através do levantamento de requisitos e analise do sistema será possível delimitar os módulos especificando suas funcionalidades.

5.7. Estruturação do programa computacional

As imagens obtidas por fotografia aérea foram lidas através de *scanners* , assim como imagens de fotografia aérea em formatado digital e georreferenciada e posteriormente armazenadas no banco de dados, utilizando técnicas de leitura e armazenagem de dados e informações. Com as imagens armazenadas na base de dados Figura 43, foi possível sua recuperação para aplicação de técnicas como, estéreoscopia, pseudoscopia e criação de mosaico controlado. Podendo reduzir assim as distorções inerentes dos levantamentos aereofotogramétricos.

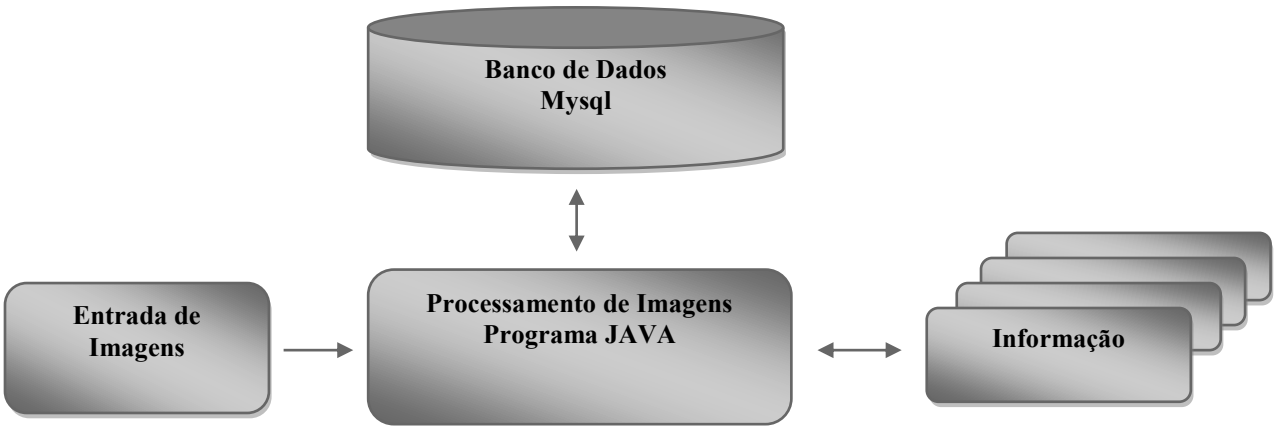

Figura 43. Armazenagem, recuperação, processamento e informação.

5.8. Desenvolvimento do programa computacional

Na fase de codificação, foi desenvolvido o núcleo básico do sistema que é transparente e pertinente à administração do sistema como um todo, abrangendo as rotinas de entrada e saída do sistema, segurança, auditoria de modificações, copia de segurança.

Na fase de codificação do núcleo básico do sistema, o esforço foi direcionado aos módulos de leitura, armazenagem e processamento das imagens.

Na fase de codificação dos módulos de geração de informações visuais.

5.8.1. Classes Java utilizadas

Para a compreensão de alguns aspectos de implementação deste projeto, fornecemos abaixo alguns detalhes relevantes das classes do Java 3D utilizadas pela aplicação.

5.8.1.1. View

O objeto View contém todos os parâmetros utilizados na renderização de uma cena tridimensional a partir de um ponto de vista. Ele contém uma lista de objetos Canvas3D nos quais esta vista é renderizada. Este objeto não faz parte do grafo de cena, porém acoplase a um nó ViewPlatform do grafo de cena. Um objeto View pode ser ativado e desativado pela aplicação em tempo de execução, sendo que a renderização de um Canvas3D só é feita quando o seu objeto View associado está ativo. Este objeto também permite iniciar e interromper o agendamento de *Behaviors*, efetivamente interrompendo a atualização de animações e interações neste objeto. Estas duas capacidades são exploradas neste projeto (Sun Microsystems, 2003).

5.8.1.2. SimpleUniverse

A classe SimpleUniverse é uma classe utilitária que automatiza a montagem do subgrafo de visualização associado a um *VirtualUniverse*. A instanciação de um objeto SimpleUniverse estabelece a configuração de um ambiente mínimo para executar um programa Java 3D, fornecendo as funcionalidades necessárias para a maioria das aplicações. Quando uma instância de SimpleUniverse é criada, ela gera automaticamente os objetos que compõem o subgrafo de visualização, como *Locale*, *ViewPlatform* e *Viewer* (Sun Microsystems, 2003).

5.8.1.3. Appearance

O objeto Appearance define os parâmetros de renderização que podem ser especificados como componentes de um nó *Shape3D*. Este parâmetros incluem: coloração, transparência, textura, entre outros. Uma subclasse importante de Appearance para este projeto é a classe *ShaderAppearance*, que permite a aplicação de um *shader*

programável ao seu objeto *Shape3D associado*. A hierarquia de classes que permite o uso de shaders no Java 3D é explicada em detalhes mais adiante (Sun Microsystems, 2003).

5.8.1.4. Canvas3D

Canvas3D é uma classe bastante complexa, pois é ela que implementa a renderização da cena em tela, possuindo diversos modos de operação e possibilidades de configuração. Por sua importância para este projeto, explicamos detalhadamente o funcionamento desta classe abaixo.

O Canvas3D é uma extensão da classe AWT Canvas, fornecendo uma área retangular na tela para a renderização tridimensional de imagens e para a captura de eventos de entrada. Pode ser renderizado em modo monoscópico ou estereoscópico. No modo monoscópico o Canvas3D pode gerar três tipos de vista:

1) View.LEFT_EYE_VIEW, a vista do olho esquerdo;
2) View.RIGHT_EYE_VIEW, a vista do olho direito;
3) View.CYCLOPEAN_EYE_VIEW, a vista correspondente ao ponto médio entre os olhos.

Estas políticas são selecionadas utilizando os métodos *setMonoscopicViewPolicy()* e *getMonoscopicViewPolicy().*

Este projeto utilizou as duas primeiras políticas de geração da vista do modo monoscópico para obter as imagens esquerda e direita, que seriam posteriormente coloridas e somadas, resultando em um anaglifo. O modo estereoscópico pode ser utilizado em sistemas que possuam suporte de hardware para tal, como sistemas com óculos obturadores (Sun Microsystems, 2000).

O Canvas3D pode ser configurado para funcionar no modo *onscreen*, no qual o seu *loop* de renderização é executado continuamente, renderizando a imagem na tela (desde que o mesmo esteja associado a um objeto View ativo), ou *offscreen*, no qual a renderização é feita em um buffer de memória em resposta a uma chamada do método renderOffscreenBuffer() (Sun Microsystems, 2003).

6 RESULTADOS E DISCUSSÕES

Temos como apresentação um programa de computador desenvolvido e nessa etapa será abordada suas funcionalidades;

Foi desenvolvido um aplicação para computador composta por uma tela inicial para o cadastro do projeto onde posteriormente poderá ser recuperado Figura 44;

Figura 44. Tela Inicial, Menu de Projetos.

Na Figura 45, obsevar-se a tela onde será pesquisado o projeto previamente cadastrado;

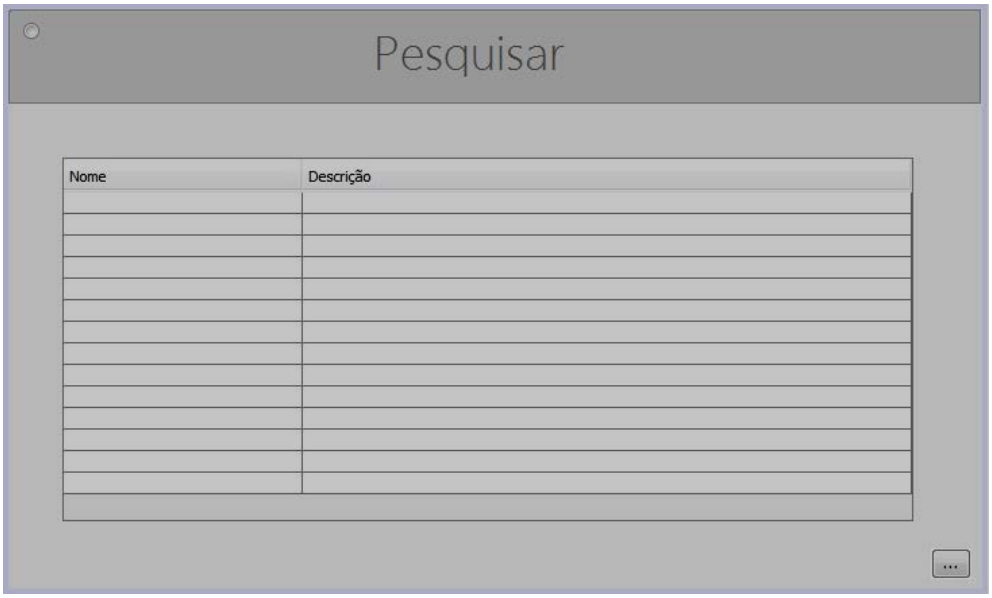

Figura 45. Tela de pesquisa de projetos.

Após a pesquisa de projetos previamente cadastrados, temos a tela de processamento, interação com as imagens obtidas por fotografia aérea previamente digitalizadas Figura 46;

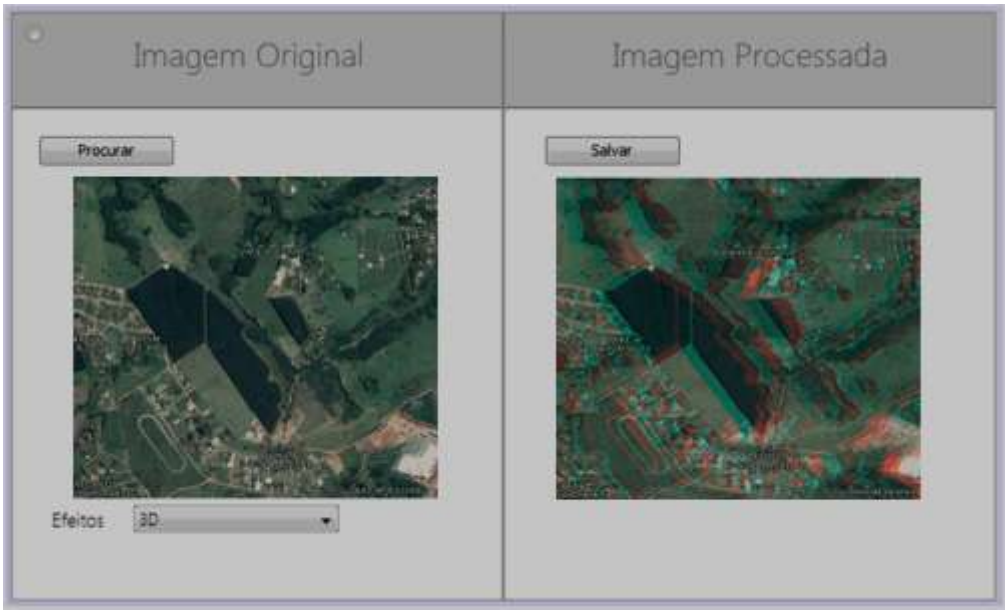

Figura 46. Tela de processamento das imagens.

Na Figura 46, observar-se a imagem original e a imagem processada, onde podemos obter informações que sem o processamento da imagem não seria possível sem uma investigação mais aprofundada por outras técnicas , como por exemplo a

estéreoscopia. Após o processamento da imagem bidimensional temos como resultado a imagem tridimensional (anaglifo).

Avaliação do Efeito Estereoscópico

A avaliação do efeito estereoscópico foi feita através da exibição do sistema em operação aos usuários de teste, por um período de 1 a 3 minutos. 30 usuários foram submetidos ao teste do programa de computador, sendo que dos usuários entrevistados:

1) A maior parte dos usuários (60%) declarou ter percebido imediatamente a sensação de profundidade causada pelo efeito estéreo;
2) 25% declararam perceber o efeito estereoscópico após um período de acomodação da visão;
3) 15% não conseguiram perceber o efeito;

Todos os usuários declararam ter percebido algum grau de persistência de cores na visão após retirar os óculos;

.

7 CONCLUSÕES.

De acordo com a metodologia proposta e os resultados obtidos com o aplicativo de computador desenvolvido concluiu-se que:

Criação do efeito estereoscópico

O objetivo principal do aplicativo consistiu em obter a renderização estereoscópica de uma imagem bidimensional obtida por meio de fotografia aérea. A aplicativo conseguiu por meio do processamento de imagem bidimensional gerar imagem utilizando anaglifos para criar o efeito estéreo. Além de renderizar a imagem em anaglifo puro, o sistema possui ainda outros 4 modos de renderização do anaglifo, bem como a capacidade de ajustar a separação interocular do observador para obter o melhor efeito.

Interatividade

A interatividade é outro requisito principal das aplicações a que este projeto se destina, e foi mantida parcialmente na versão final do programa. O problema encontrado neste aspecto foi a captura dos eventos de entrada pelo programa, que necessita de outra janela da aplicação além da janela de exibição do anaglifo.

Aplicabilidade do programa

A aplicação do programa de computador desenvolvido visa reforçar os resultado obtido por meio do uso convencional do estereoscopio na obtenção de dados e informações.

Onde estes dados podem ser confrontados com o intuito de obter maior segurança na avaliação destes e consequentemente a tomada de decisão.

8 REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, A. L. P. **Um modelo paravisualização estereoscópica utilizando webcams.** Rio de Janeiro: PUC Rio Pontifícia Universidade Católica do Rio de Janeiro, 2006.

ANDERSON, P. S**. Fundamentos para fotointerpretação.** Rio de Janeiro: Sociedade Brasileira de Cartografia, 1982.

AZUMA, R. T. **A Survey of Augmented Reality**. Malibu, CA, Hughes Research Laboratories: Artigo Científico, 1997.

AZUMA. R. **Autocalibration of an Electronic Compass in an Outdoor Augmented Reality System.** Munique, Alemanha: International Symposium on Augmented Reality 2000.

BARBOSA, M. P.; LOPES, M. A.; ZAMBALDE, A. L. **Programa computacional para gerenciamento de rebanhos bovinos: desenvolvimento e avaliação pela softhouse.** Viçosa, MG: Revista Brasileira de Agroinformática, v. 3, n. 1, p. 14, 2000.

BERTOZZI, M.; BROGGI, A. **GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection**. [s.l.], IEEE Transactions on Image Processing, 1997. URL http://citeseer.nj.nec.com/bertozzi98gold.html.

BOEMO, D. **Desenvolvimento de sistemas computacionais móveis, integrados a receptores GPS Bluetooth, aplicáveis a gestão rural e urbana.** Santa Maria: 2007. 79p. Dissertação (Mestrado em Geomática) Universidade Federal de Santa Maria, 2007.

BURGER, W.; BURGE, M. J. (2009). **Principles of Digital Image Processing: Fundamental Techniques**. [s.l.] Springer.

CAMARA, G.; MEDEIROS, J.S. Operações de análise geográfica. In: ASSAD D. E; SANO E. E. **Sistema de informações geográficas**. Brasília: Aplicação na agricultura**.** 2.ed., EMBRAPASPI/EMBRAPA-CPAC, 1998.

CERVO, A. L.; BERVIAN, P. A. **Metodologia científica**. São Paulo: Makron Books, 1996.

CRUZ, L. B. S. **Diagnóstico Ambiental da Bacia Hidrográfica do Rio Uberara - MG.** São Paulo, Campinas: Dissertação de Mestrado- Universidade Estadual de Campinas, Faculdade de Engenharia Agrícola.

DATE, C.J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2000. 803 p.

DEITEL, H.; DEITEL, P.[S.L.] (2005). Java: Como Programar. Prentice- Hall, 6th edition.

ESTEVES, G. R. P. **Estereoscopia no cálculo de distância e controle de plataforma robótica.** Universidade de Pernambuco: Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia Eletrônica pela Escola Politécnica de Pernambuco. 2011.

FERNANDES, A. R. Tópicos Avançados de Computação Gráfica GLSL Programação de Shaders.[S.L] 2006.

GIL, A. C. **Técnicas de pesquisa em economia e elaboração de monografias**. São Paulo: 4 ed. Atlas, 2002.

GONZALEZ, R. C.; WOODS, R. E. **Processamanto de Imagens Digitais.** [S.L.] Editora Edgard Blücher, 2000.

MANSSOUR, I. H. **Introdução a Java 3D.** Porto Alegre: Faculdade de Informática PUCRS, 2003.

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens.** Rio de Janeiro: Brasport Livros e Multimídia, 1999.

MILGRAM, P. et al. **Augmented Reality: A class of displays on the realityvirtuality continuum.** [S.L.] Telemanipulator and Telepresence Technologies, SPIE V.2351., 1994.

MEIRA, C. A. A. et al. **Agroinformática: Qualidade e produtividade na agricultura.** Brasília: Caderno de Ciência e Tecnologia, v. 13, n. 2, p. 175-194, 1996.

MIANO, J. (1999**). Compressed Image File Formats.** [S.L.] ACM Press - Addison-Wesley, Reading, MA.

MOREIRA, M. A**. Fundamentos do sensoriamento remoto e metodologia de aplicação.** Viçosa: 2. ed. Ed. da Universidade Federal de Viçosa, 2003. 307p.

MURRAY, L.; VANRYPER, W. (1996).[S.L.] Encyclopedia of Graphics File Formats. O'Reilly, 2nd edition. Sebastopol, CA.

NOLAN, R. L., **Management Accounting and Control of Data Processing.** [S.L.] National Association of Accountants, 1977.

RAPOSO, A.B et al. **Visão Estereoscópica, Realidade Virtual, Realidade Aumentada e Colaboração**, [S.L.] 2004.

RASKAR, R.; WELCH, G.; FUCHS, H. **Spatially Augmented Reality.** San Francisco: First International Workshop on Augmented Reality, 1998.

RODRIGUES, L. H. (2001). **Building Imaging Applications with Java Technology: Using AWT Imaging, Java 2D, and Java Advanced Imaging (JAI)**. [S.L.] Addison-Wesley Professional.

SANTOS, R. (2004). **Introdução à Programação Orientada a Objetos Usando Java.** [S.L.] Campus.

SEGAL, M.; AKELEY, K. **The Design of the OpenGL Graphics Interface.** [S.L.] Mountain View: Silicon Graphics Computer Systems, 2006.

SEVO, D. **Becoming a Computer Animator.** 01 de 01 de 2005. http://www.danielsevo.com/ (acesso em: 14 ago. 2013).

SPRAGUE, Ralph H. e HUGH J.Watson, **Sistemas de Apoio à Decisão.** Campus. 1991.

STATE, A. et al. **Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking.** Chapel Hill: Department of Computer Science University of North Carolina, 1997.

TEIXEIRA, A. L. A.; MORETTI, E.; CHRISTOFOLETTI, A. **Introdução aos sistemas de informações geográficas.** Rio Claro, São Paulo: Ed. do autor, 1992. 80p.

TORI, R.; KIRNER, C.; SISCOUTTO, R. **Fundamentos e tecnologia de realidade virtual e aumentada.** Porto Alegre: 1 ed. SBC, 2006.