UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

RETIFICAÇÃO CILÍNDRICA: UM MÉTODO EFICIENTE PARA A RETIFICAÇÃO DE PAR DE IMAGENS

TRABALHO DE GRADUAÇÃO

ALUNO **RAFAEL ROBERTO**

ORIENTADORA **VERONICA TEICHRIEB**

22 DE JUNHO DE 2009

Resumo

Com diversas possibilidades de aplicação em áreas distintas, as técnicas de reconstrução 3D, onde um mundo virtual é modelado automaticamente a partir de imagens capturadas de cenas reais, tem sido bastante estudadas e desenvolvidas nos últimos anos. Com este tipo de tecnologia é possível, por exemplo, visualizar um órgão inteiro, de qualquer ângulo, apenas a partir de imagens de radiografias, facilitando assim o diagnóstico de doenças e o modo como elas são tratadas.

As técnicas atuais que tornam esta tecnologia possível baseiam-se numa série de atividades bem estabelecidas. Nelas, as imagens capturadas são analisadas de modo que os pontos de alto contraste, aqueles que se destacam bastante dos demais ao seu redor, são identificados. Essas *features*, como são chamados, são rastreadas nas imagens seguintes e, a partir do modo como estes pontos correspondentes se relacionam pode-se encontrar a localização espacial deles. Em seguida, uma malha em três dimensões é gerada a partir desses pontos, recriando espacialmente os objetos contidos na sequência de imagens. Como nem todos os pontos de uma imagem destoam bastante dos demais à sua volta, poucas *features* são selecionadas e, consequentemente, a localização espacial de poucos pontos é encontrada, dificultando a geração do modelo tridimensional. Para evitar este problema, uma reconstrução densa é feita.

Também chamada de *dense matching*, a reconstrução densa faz uso das informações adquiridas em todo o processo de reconstrução para achar a correspondência entre todos os pontos das imagens. Para cada um deles é feita uma busca numa região específica de modo a encontrar a localização de seu correspondente nas imagens seguintes. Depois que se têm essas relações, a localização espacial da imagem inteira é encontrada. Para facilitar essa busca, as imagens são retificadas de forma que os pontos correspondentes entre uma imagem e outra estejam na mesma altura (mesma coordenada-y) e

a região de procura se resume, assim, a uma linha paralela ao eixo-x da imagem.

Uma forma simples e relativamente eficiente de retificar um par de imagens é projetando ambas num plano específico em comum. A retificação planar, como é chamada, no entanto, não é robusta o suficiente para retificar qualquer tipo de movimento de câmera. Uma alternativa é a retificação cilíndrica, que apesar de ser mais complexa, consegue englobar qualquer movimento de câmera. Nesta técnica, transformações lineares levam as imagens do sistema de coordenadas cartesiano para um sistema de coordenadas cilíndrico, mapeando cada ponto da imagem num valor (r, θ) pertencente à superfície do cilindro. No final, os valores (r, θ) são coordenadas da imagem retificada.

A retificação cilíndrica simplificada foi implementada neste trabalho e obteve bons resultados em todos os estudos de caso realizados. Na comparação com a técnica planar, esta foi mais rápida, porém a forma cilíndrica foi bem mais precisa e robusta.

AGRADECIMENTOS

Fazer um curso universitário é embarcar numa jornada de cinco anos repleta de desafios grandiosos e sacrifícios enormes. Nela a pessoa desenvolve um conhecimento técnico estrondoso, em geral, da forma mais dura possível. Afinal, é como disse o professor Sérgio Cavalcante: "só existem dois meios de um aluno aprender. Um é ele recebendo dinheiro para isso e outro é tomando porrada. Como eu não vou pagar para vocês..." *Chegar ao final da jornada exige muita determinação e gosto pelo que se faz, senão o trajeto nunca será completado.*

Entretanto, se o único ganho dessa jornada fosse o conhecimento técnico, todo o sacrifício já valeria a pena. Mas o aprendizado vai além disso. Os desafios, as novas descobertas e o convívio com outros aventureiros, cada um com suas idéias e valores, fazem com que a pessoa repense constantemente os seus atos, sua forma de ver o mundo e seu jeito de interagir com a sociedade. Parafraseando o famoso comercial, esta experiência não tem preço.

"All that you touch
All that you see
All that you taste
All you feel.
[...]
All you create
All you destroy
All that you do
All that you say.
All that you eat
And everyone you meet
All that you slight
And everyone you fight.
All that is now
All that is gone
All that's to come
And everything under the sun is in tune
But the sun is eclipsed by the moon."

- Eclipse, de Pink Floyd

Muitos dizem que sobreviveram à universidade. Confesso que em algumas situações eu também, mas no geral eu posso afirmar que eu vivi ela. Que eu aproveitei a oportunidade que me foi dada pela minha família, em especial pelos meus pais, ***Agamenon*** *e* ***Jane Flávia****, de, mesmo antes de passar no vestibular, poder me dedicar integralmente para conseguir iniciar esse caminho. A eles, eu serei eternamente grato por isso, e muito mais, assim como a outras pessoas que também contribuíram no meu preparativo como aluno e pessoa, como* ***Aline****,* ***Flávio****,* ***Glayce****,* ***Jane Félix*** *e* ***Junior****. Sem eles, talvez a jornada nunca tivesse iniciada.*

A primeira lição para a vida aprendida já nos primeiros dias é a de que a pessoa tem que ser auto-didata. Que se você não conseguir aprender sozinho, não conseguirá completar bem essa jornada. Mas, assim como o ser humano, a universidade também tem o direito de ser contraditória. Enquanto ela ensina isso,

também diz que sem ajuda dos outros que estão no mesmo barco, você também não chegará ao final. Possivelmente essas foram as maiores lições que eu tive. Eu aprendi a ser auto-didata, e desenvolvi esta técnica com ***Age****, o maior mestre de si mesmo que eu já conheci. E eu vivi essa excelente e gratificante cooperação entre pessoas, entre amigos. Sem os estudos em conjunto com* ***Jesus****,* ***Marcelo****,* ***Pyetro*** *e* ***Renata*** *talvez eu nunca tivesse saído do CTG. Sem o trabalho em equipe, dia e noite, literalmente, junto de pessoas como* ***Guilherme****,* ***Jamaj****,* ***Josias****,* ***Renato*** *e* ***Taíse****, apenas para citar alguns, muitos dos projetos nunca teriam sido concluídos.*

Mas como nem todo conhecimento é conseguido apenas através de aulas, provas e projetos, essa jornada pode ser ainda mais proveitosa dando outras oportunidades, que eu também pude aproveitar graças ao professor ***Silvio Melo****. Nelas eu pude repassar meus conhecimentos através de monitoria e enriquecer a minha base através de pesquisas científicas.*

Esse caminho também abre portas que dão acesso a outros mundos, com visões e atitudes diferentes, porém complementares, como as abertas para mim pelas professoras ***Judith Kelner*** *e* ***Veronica Teichrieb*** *do GPRT/GRVM. Elas, junto com todos os meus companheiros de trabalho, em especial a minha equipe de projeto, com* ***Andréa****,* ***Chico****,* ***Daliton****,* ***Juliane****,* ***Márcio*** *e* ***Mozart****, têm contribuído enormemente para que eu tenha acesso e também viva essas visões e atitudes tão importantes na vida de uma pessoa.*

Essa experiência é tão ampla que influencia até pessoas que não embarcaram na jornada diretamente. Às vezes eu também precisei sobreviver para conseguir concluir algumas etapas. E tenho a certeza de que só consegui suportar esses momentos graças à ajuda de pessoas importantíssimas que foram arrastadas para este caminho, como ***Carol****,* ***Ívina****,* ***Neto****, e* ***Thaís*** *durante todo o percurso e a* ***Manú****, que* "me pegou pela mão, me guiou até o final e me ajudou a entender o máximo que eu pude".

A todos vocês e a muitos outros que eu não pude citar, muito obrigado por me ajudar a viver essa experiência.

SUMÁRIO

ÍNDICE DE FIGURAS **8**

1. INTRODUÇÃO **10**

1.1. OBJETIVOS 10

1.2. ESTRUTURA DO DOCUMENTO 11

2. CONCEITOS MATEMÁTICOS **12**

2.1. GEOMETRIA PROJETIVA 12

a. Projeção ortográfica 12

b. Projeção perspectiva 13

2.2. MODELO DE CÂMERA 14

a. Parâmetros extrínsecos 16

b. Parâmetros intrínsecos 17

c. Matriz de projeção 17

2.3. GEOMETRIA EPIPOLAR 18

3. RECONSTRUÇÃO 3D A PARTIR DE IMAGENS **23**

3.1. AQUISIÇÃO DAS IMAGENS 25

3.2. SELEÇÃO DE *FEATURES* 26

3.3. CORRESPONDÊNCIA E RASTREAMENTO DE *FEATURES* 28

3.4. RECONSTRUÇÃO PROJETIVA 30

3.5. RECONSTRUÇÃO MÉTRICA 31

3.6. RECONSTRUÇÃO DENSA 32

3.7. GERAÇÃO DE MALHA E TEXTURIZAÇÃO 34

4. RETIFICAÇÃO DE IMAGENS **36**

4.1. RETIFICAÇÃO PLANAR 37

4.2. RETIFICAÇÃO CILÍNDRICA 40

4.3. RETIFICAÇÃO CILÍNDRICA SIMPLIFICADA 44

4.4. IMPLEMENTAÇÃO 48

5. RESULTADOS **50**

5.1. MOVE HOUSE 51

5.2. TEMPLE 53

5.3. AGAMENON 55
5.4. DESKTOP 57

6. CONCLUSÕES 60

6.1. TRABALHOS FUTUROS 61

7. REFERÊNCIAS BIBLIOGRÁFICAS 63

ÍNDICE DE FIGURAS

FIGURA 1: ILUSTRAÇÃO DA PROJEÇÃO ORTOGONAL 12
FIGURA 2: ILUSTRAÇÃO DA PROJEÇÃO PERSPECTIVA 13
FIGURA 3: FUNCIONAMENTO DA CÂMERA *PINHOLE* 14
FIGURA 4: MODELO DA CÂMERA *PINHOLE* 15
FIGURA 5: SISTEMA O_{XYZ} DE COORDENADAS DE MUNDO E O $O'_{X'Y'Z'}$ DE COORDENADAS DE CÂMERA 16
FIGURA 6: RELAÇÃO ENTRE OS SISTEMAS DE COORDENADAS DE CÂMERA E DE IMAGEM 17
FIGURA 7: MODELO COMPUTACIONAL PARA VISÃO ESTÉREO 19
FIGURA 8: PRINCIPAIS ELEMENTOS E RELAÇÕES DA GEOMETRIA EPIPOLAR 20
FIGURA 9: VÁRIOS PLANOS EPIPOLARES, TODOS PASSANDO PELA *BASELINE*, FORMANDO VÁRIAS LINHAS EPIPOLARES, TODAS PASSANDO PELOS EPIPOLOS 20
FIGURA 10: REPRESENTAÇÃO DO MOVIMENTO DAS CÂMERAS EM SEQUÊNCIAS COM *SHORT* E *WIDE BASELINE* 26
FIGURA 11: ORGANIZAÇÃO DOS QUADROS EM UMA SEQUÊNCIA DE IMAGENS USANDO *WIDE BASELINE* 29
FIGURA 12: UMA DAS POSSÍVEIS RECONSTRUÇÕES PROJETIVAS PARA A IMAGEM 31
FIGURA 13: RECONSTRUÇÃO MÉTRICA PARA A IMAGEM 32
FIGURA 14: ACIMA, AS IMAGENS ORIGINAIS. ABAIXO, AS MESMAS IMAGENS APÓS A RETIFICAÇÃO, COM OS PONTOS CORRESPONDENTES NA MESMA COORDENADA-Y 33
FIGURA 15: RESULTADO DA ETAPA DE RECONSTRUÇÃO DENSA 34
FIGURA 16: RESULTADO DA TRIANGULAÇÃO E TEXTURIZAÇÃO DA RECONSTRUÇÃO DENSA OBTIDA NA FIGURA 15 35
FIGURA 17: À ESQUERDA, A REPRESENTAÇÃO DE UMA IMAGEM; À DIREITA, A REPRESENTAÇÃO DE UMA IMAGEM RETIFICADA 37
FIGURA 18: ACIMA, UM PAR DE IMAGENS. ABAIXO, O MESMO PAR, PORÉM RETIFICADO, COM PLANOS DE IMAGEM CO-PLANARES, EPIPOLOS NO INFINITO E LINHAS EPIPOLARES PARALELAS 38
FIGURA 19: CASO ONDE A CÂMERA SE DESLOCOU PARA FRENTE, FAZENDO COM QUE A PROJEÇÃO DO PONTO M, NA *BASELINE*, NUNCA INTERCEPTE O PLANO DE PROJEÇÃO 40
FIGURA 20: SEQUÊNCIA DE ATIVIDADES DA RETIFICAÇÃO CILÍNDRICA 43

FIGURA 21: REPRESENTAÇÃO DA RETIFICAÇÃO CILÍNDRICA SIMPLIFICADA PARA UMA IMAGEM 44
FIGURA 22: REGIÕES ONDE O EPIPOLO PODE ESTAR LOCALIZADO 45
FIGURA 23: LINHAS EPIPOLARES EXTREMAS E REGIÃO EM COMUM 46
FIGURA 24: MENOR ÂNGULO ENTRE DUAS LINHAS CONSECUTIVAS 47
FIGURA 25: PAR ORIGINAL DA SEQUÊNCIA CONHECIDA COMO "*MOVE HOUSE*" 51
FIGURA 26: RETIFICAÇÃO PLANAR DA CENA "*MOVE HOUSE*" 52
FIGURA 27: RETIFICAÇÃO CILÍNDRICA SIMPLIFICADA DA CENA "*MOVE HOUSE*" 52
FIGURA 28: PAR ORIGINAL DA SEQUÊNCIA CONHECIDA COMO "*TEMPLE*" 54
FIGURA 29: RETIFICAÇÃO PLANAR DA CENA "*TEMPLE*" 54
FIGURA 30: RETIFICAÇÃO CILÍNDRICA SIMPLIFICADA DA CENA "*TEMPLE*" 55
FIGURA 31: PAR ORIGINAL DA SEQUÊNCIA CONHECIDA COMO "AGAMENON" 56
FIGURA 32: RETIFICAÇÃO PLANAR DA CENA "AGAMENON" 56
FIGURA 33: RETIFICAÇÃO CILÍNDRICA SIMPLIFICADA DA CENA "AGAMENON" 57
FIGURA 34: PAR ORIGINAL DA SEQUENCIA CONHECIDA COMO "*DESKTOP*" 58
FIGURA 35: RETIFICAÇÃO CILÍNDRICA SIMPLIFICADA DA CENA "*DESKTOP*" 58

1. INTRODUÇÃO

A reconstrução de objetos em três dimensões a partir de uma sequência de imagens é um campo de estudo bastante importante da área de visão computacional. O principal objetivo é melhorar a qualidade da visualização dessas imagens, possibilitando ao usuário observar com detalhes vários aspectos do objeto reconstruído, já que a sequência de imagens 2D é vista como uma cena 3D única, e não como um conjunto de imagens separadas.

Um exemplo que ilustra esta melhoria na forma de ver uma cena é a ferramenta Photosynth [1], que analisa um conjunto de imagens representando um determinado local e esse ambiente é montado a partir da colagem de uma foto com outras. Desta forma, é possível explorar esse local navegando através das fotos.

Existem várias formas de realizar a reconstrução 3D. Uma delas é baseada em uma classe de técnicas óticas e passivas chamada *Structure from Motion* (SfM), usada como base para este trabalho de graduação. A maioria das técnicas SfM estudadas e desenvolvidas na literatura possuem algumas características em comum [2]. Uma delas é a realização, num primeiro momento, de uma reconstrução com poucos pontos, de forma a conhecer algumas informações sobre a cena a ser reconstruída, para em seguida realizar uma reconstrução densa com muitos pontos, fazendo uso desse conhecimento obtido previamente. Essa reconstrução densa é beneficiada por um processamento de imagens que deixa pontos correspondentes, entre um par de figuras, com a mesma coordenada-y e é chamado de retificação.

1.1. OBJETIVOS

Neste contexto, este trabalho de graduação tem por objetivo estudar profundamente e implementar a técnica de retificação cilíndrica de imagens, baseada na descrita por Pollefeys [3]. Este método de manipulação de imagens permite a realização de uma reconstrução densa eficiente, e será mostrado ao longo do trabalho como ela se comporta manipulando imagens

com diferentes níveis de complexidade, como por exemplo, em casos onde o epipolo está dentro da imagem. Ela foi escolhida com base em observações realizadas durante um trabalho de análise das principais técnicas de retificação de imagens aplicadas em reconstrução 3D.

1.2. ESTRUTURA DO DOCUMENTO

O presente texto foi escrito seguindo a seguinte estrutura: no capítulo 2 serão mostrados os conceitos matemáticos básicos para o entendimento da técnica de retificação de imagens. No capítulo 3 será explicado como realizar uma reconstrução 3D a partir de uma sequência de imagens 2D, utilizando a técnica SfM. As principais formas de retificar uma imagem serão elucidadas no capítulo 4, para numa segunda parte do texto explanar com mais detalhe a técnica focada neste trabalho. Desta feita, no capítulo 5 serão expostos os resultados obtidos com a implementação da técnica. Por fim, no capítulo 6, as conclusões deste trabalho serão discutidas, assim como as possibilidades de trabalhos futuros.

2. CONCEITOS MATEMÁTICOS

Qualquer imagem, na forma como é comumente usada, consiste basicamente na projeção de um ambiente em três dimensões num plano específico. Esta foto pode ser gerada usando uma câmera, que especifica precisamente como o mundo 3D capturado vai ser representado por uma superfície 2D. Para retificar uma imagem, além de conhecer bem como essas projeções são realizadas, é necessário também saber como duas imagens diferentes da mesma cena 3D se relacionam entre si. Por isso, este capítulo apresenta os principais conceitos matemáticos que serão usados no decorrer deste trabalho de graduação.

2.1. GEOMETRIA PROJETIVA

Também chamada de geometria descritiva, esta é a área da matemática que lida com as propriedades e a consistência (no sentido de mudança de sistema de coordenadas) de figuras geométricas em relação à projeção [4]. Essas formas geométricas, em geral, são projetadas de maneira ortográfica ou perspectiva.

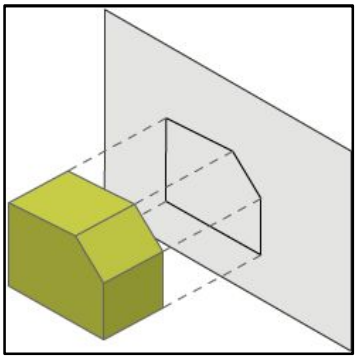

Figura 1: ilustração da projeção ortogonal

a. Projeção ortográfica

Este tipo de projeção é bastante usado em arquitetura e cartografia, por exemplo, pois preserva formas e ângulos, como pode ser visto na Figura 1. Nela pode-se observar que retas que são paralelas no objeto em três dimensões permanecerão paralelas na imagem projetada. Para produzir este efeito, os pontos do objeto 3D são levados para o plano onde estes serão

projetados de forma que as linhas de projeção que os conectam sejam ortogonais ao plano projetivo.

b. Projeção perspectiva

A projeção perspectiva é mais usada em atividades cotidianas porque ela representa os objetos no plano 2D da mesma forma que o olho humano interpreta as formas que ele enxerga. Assim, as cenas capturadas usando este tipo de projeção são mais realistas, tendo a projeção perspectiva duas importantes características que a destacam da projeção ortográfica: objetos que estão mais distantes do plano projetivo são menores do que os mais próximos e objetos aparentam ser menores do que realmente são porque possuem uma certa inclinação em relação ao plano. A Figura 2 exemplifica uma projeção perspectiva.

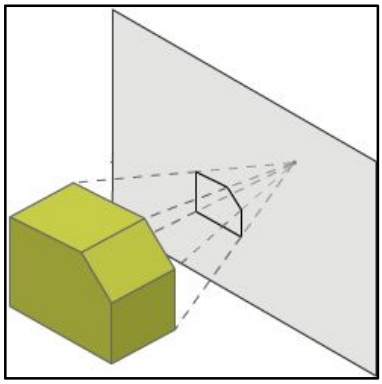

Figura 2: ilustração da projeção perspectiva

Essas características são obtidas se os objetos forem projetados direto num ponto central e desenhados no ponto onde as linhas de projeção encontram o plano projetivo.

Em geometria projetiva, pode-se definir que linhas paralelas se intersectam no infinito. Assim, é correto afirmar que a projeção ortográfica pode ser mostrada como uma projeção perspectiva onde o ponto central está numa distância infinita do plano projetivo.

Outra definição importante da geometria projetiva é a de classe de equivalência. Nela pode-se dizer que no espaço 3D, exceto a origem, dois

pontos P e P′ são equivalentes se existe um número real não nulo λ que torna a relação P = λP′ verdadeira. Assim, para se manter essa classe de equivalência e deixar as operações no espaço projetivo tão simples como no plano cartesiano, usa-se coordenadas homogêneas.

Com este tipo de coordenadas, um ponto 3D é escrito com uma dimensão a mais de tal forma que a seguinte equivalência seja preservada:

$$\begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \equiv \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \qquad (1)$$

Por convenção, define-se que os pontos com peso w = 1 estão no plano projetivo e que os pontos homogêneos com w = 0 estão no plano no infinito.

2.2. MODELO DE CÂMERA

As câmeras convencionais usam princípios de geometria projetiva, aplicando uma projeção perspectiva do objeto a ser capturado. O tipo de câmera mais simples é conhecido como *pinhole* e, no lugar de uma lente, possui uma pequena abertura, que é por onde passa a luz que irá interceptar o plano de imagem. A Figura 3 mostra como funciona este tipo de câmera.

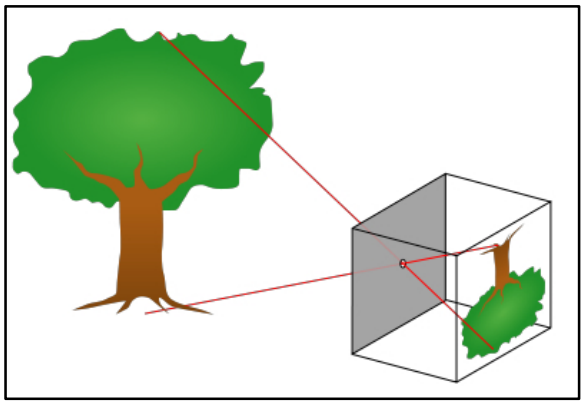

Figura 3: funcionamento da câmera *pinhole*

No contexto da computação foi definido um modelo matemático para uma câmera *pinhole* ideal. Assim como na câmera real, os raios de luz refletidos ou emitidos pelo objeto passam por um ponto chamado centro de

câmera, que representa a abertura da câmera real, e interceptam o plano de imagem, que fica a uma certa distância do centro de câmera, denominada distância focal. A reta que é perpendicular ao plano de imagem e passa pelo centro de câmera é chamada de eixo óptico. O ponto principal é o local onde este eixo intersecta o plano de imagem e, na maioria dos casos, ele representa o centro da imagem. A Figura 4 ilustra este modelo de câmera. Desta forma, um ponto M, em três dimensões, é projetado de forma perspectiva no plano de imagem resultando no ponto m, em duas dimensões.

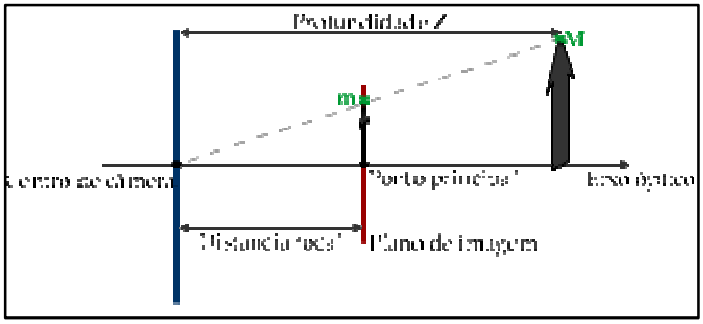

Figura 4: modelo da câmera *pinhole*

Neste modelo, o centro de câmera está localizado na origem do sistema de coordenadas de mundo e o plano de imagem é o plano Z = 1, ou seja, a distância focal é de uma unidade. Deste modo, pode-se observar que a projeção perspectiva do ponto M(x, y, z) vai resultar no ponto m(u, v) onde:

$$u = \frac{x}{z} \quad \text{e} \quad v = \frac{y}{z}. \qquad (2)$$

Entretanto, nem sempre o centro da câmera estará localizado na origem do sistema de coordenadas de mundo. Neste caso, é necessário definir um sistema de coordenadas local, onde a sua origem será o centro de câmera, de forma que as características do modelo *pinhole* sejam preservadas. Para isso, deve-se aplicar transformações lineares nos objetos de forma que eles possam ser representados no sistema de coordenadas local e projetados corretamente no plano de imagem. Essas transformações são determinadas a partir dos parâmetros intrínsecos e extrínsecos da câmera.

a. Parâmetros extrínsecos

São estes parâmetros que descrevem a posição e orientação do centro de câmera em relação ao sistema de coordenadas de mundo. Como pode ser visto na Figura 5, existe um vetor t, descrito em coordenadas de mundo, que desloca o centro de câmera O′ da origem O para uma nova posição. Na figura também pode ser observado que, além de uma posição diferente do centro de câmera, os eixos do sistema de coordenadas de câmera sofreram uma rotação R, também descrita em coordenadas de mundo, em relação à disposição original do sistema de coordenadas mundial.

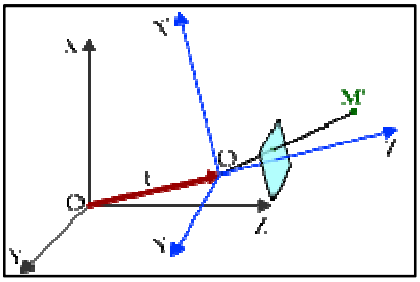

Figura 5: sistema O_{XYZ} de coordenadas de mundo e o $O'_{X'Y'Z'}$ de coordenadas de câmera

Com estes parâmetros de rotação e translação é possível escrever um ponto M′, em coordenadas de câmera, como o ponto M, em coordenadas de mundo. Para isso, é necessário aplicar ao ponto M′ as mesmas transformações aplicadas à câmera, ou seja:

$$M=\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}.M' . \qquad (3)$$

Deve-se notar que, como a matriz R e o vetor t estão em coordenadas de mundo, o ponto M também estará.

Na situação onde se tem o ponto M em coordenadas de mundo, o ponto M′ em coordenadas de câmera pode ser encontrado invertendo a matriz:

$$M' = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}^{-1} . M = \begin{bmatrix} R^T & -t \\ 0 & 1 \end{bmatrix} . M \,. \qquad (4)$$

b. Parâmetros intrínsecos

Os parâmetros intrínsecos mostram como um ponto em coordenadas de câmera será descrito no plano de imagem. Para isso, um novo sistema de coordenadas é definido: o sistema de coordenadas de imagem. Sua origem O_I está na borda superior esquerda da imagem e a sua unidade básica é o *pixel*.

Na Figura 6 pode-se observar que a relação entre um ponto de câmera e um *pixel* na imagem é dada por:

$$u = -k_u \cdot x' + u_0 \quad \text{e} \quad v = -k_v \cdot y' + v_0 \,, \qquad (5)$$

onde as constantes k_u e k_v são definidas pelo cociente entre o tamanho do *pixel* e a dimensão da imagem em cada sentido. O valor da divisão k_v/k_u é chamado de *aspect ratio*. Assim, usando coordenadas homogêneas e sabendo que f é a distância focal, tem-se que:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{-f}{k_u} & 0 & u_0 \\ 0 & \frac{-f}{k_v} & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{K} . \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} . \qquad (6)$$

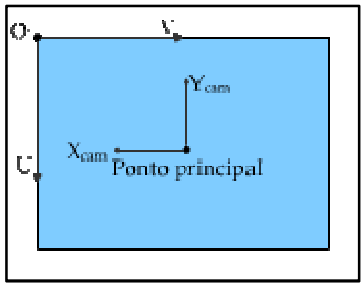

Figura 6: relação entre os sistemas de coordenadas de câmera e de imagem

c. Matriz de projeção

Tendo calculado os parâmetros intrínsecos e extrínsecos, pode-se determinar uma matriz que projeta qualquer ponto M em coordenadas de mundo num ponto m em coordenadas de imagem. Esta matriz de projeção é formada,

primeiramente, pela transformação que leva um ponto em coordenadas de mundo para coordenadas de câmera usando os parâmetros extrínsecos. Em seguida, uma projeção perspectiva é aplicada. Por fim, usa-se os parâmetros intrínsecos para encontrar as coordenadas de imagem do ponto. A composição dessas transformações resulta na seguinte equação:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{-f}{ku} & 0 & u_0 \\ 0 & \frac{-f}{kv} & v_0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{z'} & 0 \end{bmatrix} . \begin{bmatrix} R^T & -t \\ 0 & 1 \end{bmatrix}}_{P} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = K.\left[R^T \middle| -t\right] . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} . \qquad (7)$$

2.3. GEOMETRIA EPIPOLAR

Os seres humanos, de uma forma geral, possuem uma boa noção de profundidade. Nós conseguimos distinguir bem que determinados objetos estão mais próximos que outros quando olhamos para eles. Isto se deve ao fato de possuirmos visão estéreo; ou seja, cada um dos nossos olhos observa o mundo de pontos de vista diferentes e, a partir disto, o nosso cérebro consegue extrair várias relações geométricas entre as imagens formadas em cada retina e, assim, reconstrói o ambiente 3D de forma que possamos perceber as diferenças de profundidade dos objetos que compõem o ambiente.

A simulação computacional deste processo de visão para recriar o ambiente 3D a partir de duas imagens é baseada na área da geometria chamada geometria epipolar. Ela depende apenas dos parâmetros de câmera, independente da estrutura da cena [3].

A modelagem da visão estéreo pode ser realizada usando duas câmeras *pinhole*, como mostrado na Figura 7. Pode-se observar que cada câmera possui o seu próprio centro e sua própria orientação. Deste modo, cada uma possui também o seu próprio sistema de coordenadas de câmera.

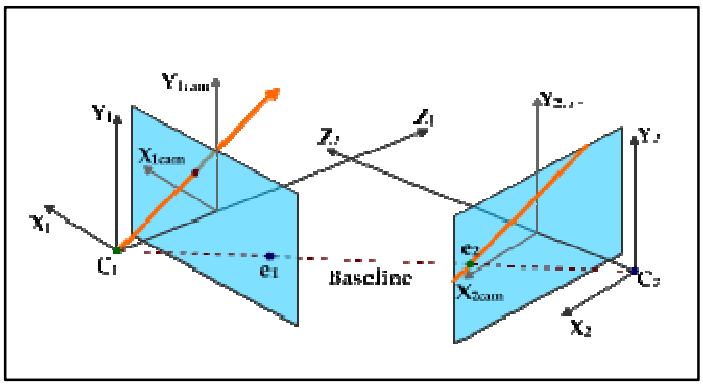

Figura 7: modelo computacional para visão estéreo

Dentre as várias relações possíveis entre pares de imagens, algumas são bastante importantes, pois ocorrem em todos os casos de visão estéreo. A primeira delas é a reta que liga o centro C_1 da primeira câmera com o centro C_2 da segunda, chamada de *baseline*. O ponto de intersecção desta reta com o plano de imagem é chamado de epipolo. Para a primeira câmera têm-se o epipolo e_1 e para a segunda câmera tem-se o epipolo e_2.

Se existem dois pontos m_1 e m_2 na primeira e na segunda imagem, respectivamente, que são a projeção de um ponto M em coordenadas de mundo, pode-se dizer que M, C_1, C_2 são coplanares, formando o plano epipolar, como visto na Figura 8. Este plano intersecta com o plano de imagem de cada uma das câmeras formando as linhas epipolares.

A Figura 8 ilustra a relação que as linhas epipolares possuem entre uma imagem e outra. Fazendo uma análise usando o ponto m_1 como referência é possível definir um raio que parte de C_1 até m_1. A partir deste raio, pode-se perceber que m_1 na realidade não é apenas a projeção de M, mas sim de todos os pontos que pertencem ao raio. Isto significa que é impossível determinar exatamente a posição espacial de um ponto projetado numa imagem sem que haja uma outra, capturada por uma segunda câmera em uma outra posição. Neste exemplo, m_2 seria este segundo ponto de vista de P. Desta forma, a intersecção dos raios que vão de C_1 à m_1 e de C_2 à m_2 ocorreria no ponto M. Se o primeiro raio for projetado na segunda imagem ele formará uma reta no plano projetivo, que é a linha epipolar correspondente ao ponto

m_1 e esta reta contém o ponto m_2. O mesmo acontece se o raio de C_2 à m_2 for projetado na primeira imagem. Desta análise pode ser extraída mais uma importante conclusão: para todos os pontos de uma imagem, seu correspondente na outra figura estará na sua respectiva linha epipolar.

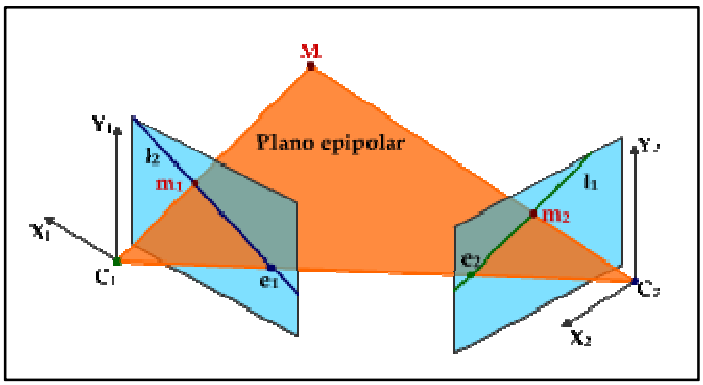

Figura 8: principais elementos e relações da geometria epipolar

Todas as linhas epipolares passam pelo epipolo da imagem e, independente da coordenada espacial do ponto M, todos os planos epipolares passarão pela *baseline*, como mostra a Figura 9.

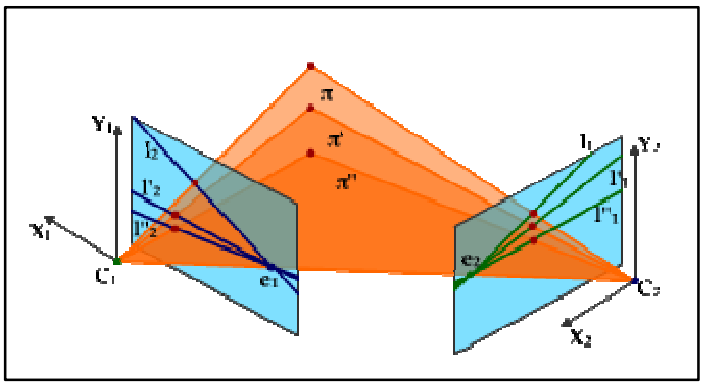

Figura 9: vários planos epipolares, todos passando pela *baseline*, formando várias linhas epipolares, todas passando pelos epipolos

A partir de um ponto m_1 na primeira imagem, a linha epipolar l_1 na segunda imagem, que irá conter o ponto m_2, pode ser achada a partir da seguinte relação:

$$l_1 = F \cdot m_1 ; \qquad (8)$$

onde F, chamada de matriz fundamental, é uma representação algébrica da

geometria epipolar entre duas imagens. Ela é uma matriz 3x3 de *rank* 2 que pode ser encontrada a partir da seguinte equação:

$$m_2^T \cdot F \cdot m_1 = 0 . \quad (9)$$

Para casos onde o objetivo é encontrar a linha epipolar l_2 na primeira imagem, correspondente ao ponto m_2 na segunda, a matriz fundamental também pode ser usada:

$$l_2 = F^T \cdot m_2 . \quad (10)$$

Outro papel importante da matriz fundamental é calcular os epipolos da imagem. Eles são definidos como os núcleos da fundamental. Ou seja:

$$F \cdot e_1 = 0 \quad \text{e} \quad F^T \cdot e_2 = 0 . \quad (11)$$

Também é possível encontrar linhas epipolares correspondentes. Ou seja, dado que a equação da linha l_1 na segunda imagem é conhecida, encontrada a partir do ponto m_1 na primeira imagem, é possível encontrar a linha epipolar l_2 que contém o ponto m_1, mesmo sem conhecer o ponto m_2 na segunda imagem. Isto é possível porque existe uma matriz de homografia H que mapeia todos os pontos e retas da primeira imagem na segunda, assim como o contrário. Esta matriz é encontrada a partir da matriz fundamental e dos epipolos da imagem pela seguinte equação:

$$H = [e_2]_X \cdot F + e_2 \cdot a^T ; \quad (12)$$

onde a é um vetor qualquer, não nulo, usado para garantir que a matriz H seja invertível e $[e_2]_X$ é a matriz anti-simétrica do epipolo da segunda imagem, definida por:

$$[e_2]_X = \begin{bmatrix} 0 & -e_2.z & e_2.y \\ e_2.z & 0 & -e_2.x \\ -e_2.y & e_2.x & 0 \end{bmatrix} . \quad (13)$$

Caso a matriz de projeção seja conhecida, a homografia também pode ser encontrada como:

$$H^{-T} = \left(P_2^{T}\right)^{+} \cdot P^{T}; \qquad (14)$$

onde $\left(P_2^{T}\right)^{+}$ é a pseudo-inversa de Moore-Penrose da transposta da matriz de projeção da segunda imagem.

Assim, conhecendo a matriz de homografia, as linhas epipolares correspondentes podem ser facilmente calculadas usando:

$$l_2 \approx H^{-T} \cdot l_1 \quad \text{e} \quad l_1 \approx H^{T} \cdot l_2 . \qquad (15)$$

3. RECONSTRUÇÃO 3D A PARTIR DE IMAGENS

Um importante tópico de pesquisa na área de visão computacional consiste em gerar modelos digitais em três dimensões automaticamente a partir de imagens bidimensionais de cenas de interesse [5]. Isso se deve à sua grande aplicabilidade em importantes atividades cotidianas. Recriar objetos vem despertando interesse de vários campos de atuação por melhorar significativamente a qualidade da visualização. Na medicina, por exemplo, a partir de uma sequência de radiografias de um tumor, o modelo do câncer e dos órgãos ao redor deste pode ser reconstruído em três dimensões, possibilitando aos médicos um diagnóstico mais eficiente [6]. Outras áreas, como realidade virtual, computação gráfica, engenharia civil e robótica, apenas para citar algumas, também estão atentas aos resultados obtidos pela reconstrução 3D [7].

Um segundo motivo para a popularização da reconstrução 3D baseada em imagens 2D se deve, principalmente, ao barateamento dos equipamentos usados para o seu desenvolvimento. Enquanto anteriormente eram necessários dispositivos especializados, como *scanner* a *laser*, que encareciam o valor do sistema, hoje pode-se obter resultados semelhantes usando câmeras de vídeo convencionais. Assim, a partir de uma sequência de imagens adquiridas movimentando livremente a câmera em torno do molde original, uma representação em escala do modelo original é gerada em 3D e, em seguida, texturizada com uma imagem da cena real de modo a aumentar o realismo da mesma. Para isso, não é necessário, obrigatoriamente, conhecer os parâmetros da câmera.

Esta técnica de reconstrução 3D a partir do movimento livre da câmera em torno do modelo original é chamada de *Structure from Motion* [3]. Também chamada de SfM, esta técnica é bastante estudada e admite várias formas de implementação, dependendo das informações disponíveis antes e durante o

processo de reconstrução, como o conhecimento prévio dos parâmetros da câmera, ou do objetivo final da aplicação.

Uma destas possíveis formas está sendo implementada no projeto TechPetro [8], desenvolvido pelo Grupo de Pesquisa em Realidade Virtual e Multimídia, GRVM [9], do Centro de Informática [10] da Universidade Federal de Pernambuco [11]. O projeto tem por objetivo, entre outros, desenvolver um *framework* de reconstrução 3D utilizando o SfM como base. O presente trabalho de graduação vem contribuir com a reconstrução densa, que é uma das etapas do processo de reconstrução 3D, como pode ser visto a seguir. O *pipeline* de reconstrução 3D baseado em SfM é descrito detalhadamente neste capítulo.

1. Aquisição das imagens: as imagens são capturadas usando uma câmera de vídeo (sequência de vídeo) ou fotográfica (sequência de fotos).

2. Seleção de *features*: as imagens possuem pontos que se destacam bastante dos *pixels* ao seu redor, sendo facilmente identificados. Essas *features*, como são chamadas, são identificadas em todas as imagens.

3. Correspondência e rastreamento de *features*: as mesmas *features* encontradas nas imagens são relacionadas entre si de forma que elas possam ser rastreadas ao longo da sequência.

4. Reconstrução projetiva: o movimento da câmera e a posição espacial das *features* são encontrados a partir de uma transformação de projeção.

5. Reconstrução métrica: com os parâmetros da câmera conhecidos, é possível descobrir uma matriz de homografia que leva as posições 3D das *features* obtidas pela reconstrução projetiva para um sistema métrico.

6. Reconstrução densa: de posse de todos os parâmetros da câmera, adquiridos no processo de reconstrução das *features*, o *dense matching*, como também é chamado, recria praticamente todos os pontos da imagem para que o modelo gerado seja o mais fiel possível à forma do modelo original.

7. Geração de malha e texturização: as informações de cor do objeto original são repassadas aos pontos reconstruídos para que a reconstrução possua também, além de uma forma semelhante, uma aparência visual idêntica.

3.1. AQUISIÇÃO DAS IMAGENS

Este primeiro passo consiste em capturar a sequência de imagens que será usada na reconstrução e depende bastante da forma como o SfM foi implementado. Para as formas mais simples do SfM é necessário que a captura de imagens seja feita de forma bem controlada. Isto significa que os parâmetros internos da câmera são conhecidos e permanecem constantes durante todo o processo de aquisição. Isto torna desnecessário o uso de algoritmos para descobrir o valor da matriz K, descrita na subseção 2.2.b. Além disso, é conveniente usar objetos que estabilizem a captura das imagens, como tripé ou trilhos, evitando movimentos bruscos e inesperados da câmera.

Porém, nem sempre é possível ter um ambiente controlado para adquirir as imagens, pois este preparo pode encarecer bastante o custo da reconstrução. Para determinadas aplicações pode ser necessário até o uso de câmeras de vídeo convencionais, com a captura feita à mão livre. Nesse caso, as imagens poderão sair tremidas, com algumas delas borradas. Além disso, muitas dessas câmeras modernas possuem um sistema de foco automático, onde as lentes são ajustadas de forma que as imagens sejam sempre nítidas. Se a distância focal varia, a matriz K também não permanece constante durante toda a captura. Para esses casos, a reconstrução tem que ser mais robusta, o que pode tornar necessário o uso de um pré-processamento das

imagens, eliminando ou tornando nítidas as imagens borradas, ou de algoritmos de auto-calibração, que deverão descobrir o valor da matriz de parâmetros intrínsecos a partir das próprias imagens.

Seja em ambientes controlados ou não, as imagens podem ser adquiridas por equipamentos diferentes, como câmeras de vídeo ou fotográficas. Para o primeiro caso, a posição da câmera ao longo da sequência de imagens tende a mudar de forma mais suave. Ou seja, figuras consecutivas do conjunto tendem a variar muito pouco. Isso faz com que a *baseline* seja curta, já que a distância do centro de projeção da primeira imagem para a segunda é muito pequena, sendo denominada de *short baseline*.

Quando uma sequência de fotos é usada, a variação entre as imagens tende a ser maior. Assim, é comum que a distância entre os centros das câmeras seja grande, caracterizando a chamada *wide baseline*. Cada um desses tipos de equipamentos tem as suas vantagens e desvantagens. Por isso, em geral, é usado um misto das duas, fazendo uma reconstrução com *wide baseline* para, em seguida, melhorá-la com *short baseline*. A Figura 10 ilustra a diferença entre as distâncias entre os centros de projeção das imagens. Nela é possível perceber que o *short baseline* possui mais representações de câmera para captura a mesma cena em comparação com *wide baseline*.

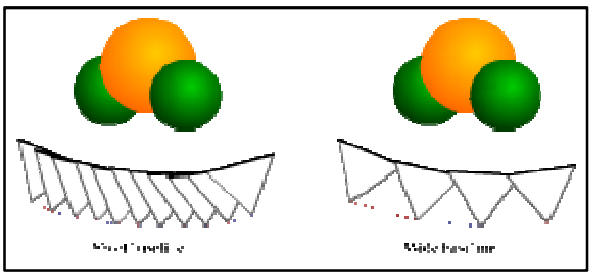

Figura 10: representação do movimento das câmeras em sequências com *short* e *wide baseline*

3.2. SELEÇÃO DE *FEATURES*

Depois que o conjunto de imagens do objeto a ser reconstruído foi obtido, cada uma delas é analisada em busca de pontos que possam ser facilmente

identificados, ou seja, que se destacam dos demais ao seu redor. Dada a imagem de uma casa, por exemplo, com uma porta pintada uniformemente da cor branca, um ponto de evidência seria a fechadura já que, diferentemente do resto da porta, ela é escura. Esses pontos de grande destaque são chamados de *features* que, por se diferenciarem de seus vizinhos, podem ser facilmente determinados na sequência de imagens. Ainda com o exemplo da porta, é mais fácil encontrar a fechadura em duas imagens distintas e saber que elas correspondem ao mesmo ponto do modelo original, do que qualquer outro ponto branco da porta, já que eles são idênticos.

Potencialmente, cada ponto m(u, v) na imagem pode ser uma *feature*. Então, para determinar precisamente quais realmente serão escolhidos, cada *pixel* é avaliado por um critério conhecido como *Harris Corner Detector*. Ele mede o quanto cada ponto de destaca dos demais. Se esse valor for maior que um limiar pré-estabelecido, então ele é considerado uma *feature* [2].

O critério de *Harris* para um ponto m qualquer na imagem é dado pela equação:

$$C(m) = \det(G) + k \times tr^2(G), \tag{16}$$

onde k é uma constante muito pequena, na ordem de 10^{-2}, pertencente aos reais. Já G representa uma matriz composta por vetores que indicam quanto os valores dos *pixels* próximos à m estão variando em relação a ele, tanto na direção x quanto na direção y. Esses vetores, chamados de gradiente, são calculados usando os valores dos pontos de uma região W(m) cujo tamanho, em geral, é 7 x 7 e o seu centro é m. Desta feita, G pode ser escrita da seguinte forma, com I_x e I_y sendo os vetores gradiente obtidos a partir da convolução da imagem I com a derivada de um par de filtros gaussianos:

$$G = \begin{bmatrix} \sum_{W(x)} I_x^2 & \sum_{W(x)} I_x I_y \\ \sum_{W(x)} I_x I_y & \sum_{W(x)} I_y^2 \end{bmatrix}. \tag{17}$$

Deste modo, o ponto m será selecionado como uma *feature* se o resultado de C(m) for maior que o limiar. Como imagens podem ter algumas áreas com mais texturas do que outras, é interessante dividi-las em regiões e determinar um limiar para cada uma dessas sub-áreas, ao invés de um único para toda a figura.

Não há um limite superior ou inferior para o número de *features* em cada região, entretanto é necessário que haja uma distância mínima entre eles. Usando mais uma vez o exemplo da porta e supondo que uma fechadura escura ocupe uma área de 5 x 5 *pixels*, se for usada uma janela W(m) de tamanho 11 x 11, cada um dos 25 pontos que compõem a fechadura vão ser considerados *features*. Para evitar problemas de rastreamento, apenas o *pixel* que tiver o maior valor é selecionado e os outros são omitidos.

3.3. CORRESPONDÊNCIA E RASTREAMENTO DE *FEATURES*

Existem, também, muitas maneiras de rastrear *features* numa sequência de imagens. Dado que elas tenham sido encontradas usando *Harris Corner Detector*, pode-se usar, por exemplo, uma técnica de correlação cruzada normalizada [12].

Nela, para cada *feature* na primeira imagem, é feita uma busca para identificar com qual ponto de destaque na segunda ela mais se assemelha. Em seguida, o mesmo é feito no sentido inverso, da segunda imagem para a primeira. Se duas *features*, em imagens distintas, forem mutuamente as mais semelhantes, então elas são consideradas correspondentes.

O problema desta técnica é que o seu tempo de execução cresce bastante a cada imagem adicionada no rastreamento, pois para cada *feature* em cada imagem é necessário fazer uma busca em todas as outras figuras da sequência, assim como a pesquisa inversa. Uma alternativa mais eficiente computacionalmente é usar o *Kanade-Lucas-Tomasi tracker* [13].

Primeiramente, para uma sequência de imagens, o KLT *tracker*, como é conhecido, detecta um conjunto de pontos de interesse, que potencialmente

são de fácil rastreamento nos quadros subseqüentes. Como vimos na seção anterior, esta detecção também pode ser feita usando o *Harris Corner Detector*, mas com um critério diferente.

Depois de calculados os vetores gradiente em cada sentido, para cada *feature*, e um auto-valor λ para ambas as orientações, é encontrada uma transformação afim que mapeia toda a vizinhança da *feature* de uma imagem na sua correspondente da outra. Essa transformação faz a correlação minimizando a falta de similaridade entre as duas regiões.

Diferentemente da correlação cruzada, onde há a necessidade de usar um *short baseline* para fazer o rastreamento das *features*, o KLT atua usando *wide baseline*. Desta forma, não há a necessidade de executar o algoritmo entre todas as imagens da sequência. Ao invés disso, pode-se escolher quadros distintos, que marcam o início e o fim de uma sequência de figuras levemente diferentes, do conjunto de imagens, chamados de *keyframes*, e fazer o rastreamento somente entre eles. A Figura 11 ilustra como é feita esta organização de quadros numa sequência de imagens.

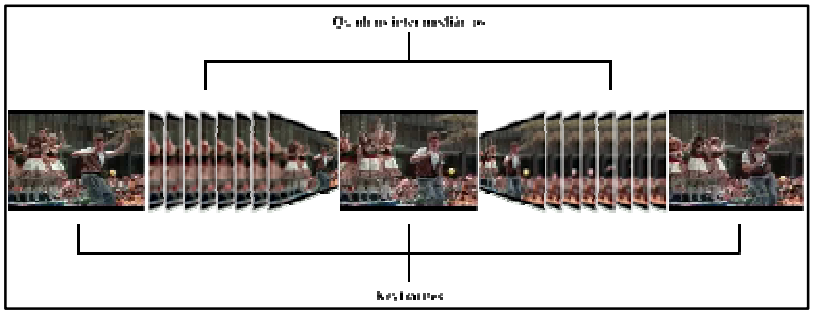

Figura 11: organização dos quadros em uma sequência de imagens usando *wide baseline*

Com esta informação sobre como as *features* se relacionam numa determinada sequência de imagens, pode-se usar esses pontos correspondentes entre dois quadros para encontrar a matriz fundamental entre pares de *keyframes* utilizando um algoritmo chamado de *eight-point*, como descrito em [14]. Depois de achar a fundamental, ela ainda pode ser melhorada usando os quadros intermediários.

3.4. RECONSTRUÇÃO PROJETIVA

De posse das *features* e da informação de como elas se relacionam na sequência de imagens, além da matriz fundamental, já é possível recuperar a estrutura 3D de uma cena. Porém, como ainda não se tem os parâmetros intrínsecos da câmera, esta reconstrução será feita a partir de uma transformação de projeção, e não a partir de uma escala. Isso se deve ao fato de que existe uma decomposição da matriz fundamental capaz de obter as matrizes de projeção de ambas imagens.

A decomposição da matriz fundamental é possível porque, como $F = [t]_x KRK^{-1}$, todas as matrizes de projeção $P = \left[KRK^{-1} + t \cdot v^T \,|\, v_4 \cdot t\right]$ geram a mesma matriz fundamental, independente dos valores do vetor $v = [v_1, v_2, v_3]$ e da constante v_4. Entretanto, para cada valor de v e v_4 haverá uma matriz de projeção diferente. Uma das formas de decompor a matriz fundamental é usar a chamada decomposição canônica, onde tenta-se fazer com que a matriz de projeção dependa apenas da fundamental, e não de v e v_4. Assim, temos:

$$P_1 = [I \,|\, 0] \quad \text{e} \quad P_2 = \left[[t]_x^T \cdot F \,|\, t\right]. \tag{18}$$

Com as matrizes de projeção é possível reconstruir um ponto M, em coordenadas de mundo, a partir de uma relação entre elas e os pontos correspondentes m_1 e m_2, em coordenadas de imagem:

$$\lambda_1 \cdot m_1 = P_1 \cdot M \quad \text{e} \quad \lambda_2 \cdot m_2 = P_2 \cdot M\,. \tag{19}$$

Para cada matriz de projeção que pode ser encontrada, um valor diferente de M também é achado e, consequentemente, várias reconstruções distintas. Todas elas estão corretas, apesar de aparentarem possuir uma distorção. Esta distorção se deve justamente a esse ajuste dos parâmetros v e v_4. A Figura 12, a seguir, mostra uma das possíveis reconstruções projetivas para uma determinada imagem.

Figura 12: uma das possíveis reconstruções projetivas para a imagem [15]

3.5. RECONSTRUÇÃO MÉTRICA

Em geral, a reconstrução projetiva não é realizada já que, ou os parâmetros intrínsecos da câmera são conhecidos no momento da captura ou eles são descobertos num processo chamado auto-calibração da câmera. Assim, de posse da matriz fundamental e das informações de câmera é possível fazer a reconstrução da cena usando a matriz essencial, que é uma matriz que descreve a relação métrica entre dois pontos correspondentes, semelhante à fundamental, que relaciona essas correlações de forma projetiva. A matriz essencial pode ser encontrada através da seguinte equação:

$$E = K_2^{T} \cdot F \cdot K_1 . \qquad (20)$$

A reconstrução é possível porque, a partir da matriz essencial pode-se determinar tanto a posição relativa como a orientação entre as diversas posições da câmera e a localização espacial dos pontos das imagens. A matriz essencial é definida como:

$$E = R \cdot [t]_x . \qquad (21)$$

Desta forma, a matriz essencial pode ser decomposta usando o *Singular Value Decomposition*, ou SVD, que é uma decomposição em valores singulares, encontrando assim as matrizes de rotação e translação, numa determinada escala, relativas entre duas imagens [16].

Como as matrizes de rotação e translação encontradas são relativas entre duas imagens e não para toda a sequência, é necessário fazer um

alinhamento de todas essas matrizes para que a escala seja a mesma na sequência inteira. Desta feita, atribui-se que a posição da câmera para a primeira imagem é a origem do sistema de coordenadas de mundo e ela não possui rotação. Usando a matriz essencial entre o primeiro quadro e o segundo, encontra-se a rotação e a translação entre eles. Posteriormente, usando a matriz essencial que leva da segunda para a terceira imagem, encontra-se as matrizes R e t entre essas duas figuras e, em seguida, essas matrizes são alinhadas com as posteriores para que fiquem na mesma escala. Este processo é repetido para todas as imagens da sequência.

No final, de posse da posição e orientação de todas as câmeras, assim como dos parâmetros intrínsecos, a matriz de projeção para cada pose é encontrada. Com ela, os pontos em coordenadas de mundo podem ser encontrados usando as mesmas relações da reconstrução projetiva. A Figura 13 ilustra o resultado da reconstrução métrica.

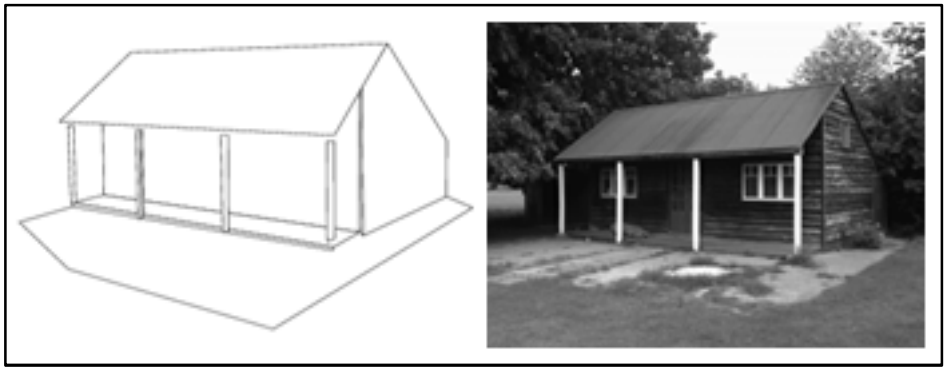

Figura 13: reconstrução métrica para a imagem [15]

3.6. Reconstrução densa

Neste estágio da execução da técnica já se possui uma reconstrução muito boa da cena, onde o modelo reconstruído consegue ser identificado. Porém, como ela foi feita baseada apenas nas *features*, existem poucos pontos para serem exibidos, fazendo com que o modelo 3D possa não ser tão detalhado. Entretanto, como já se possui todas as informações de câmera da sequência de imagens, é possível melhorar a qualidade do modelo aumentando o número de pontos reconstruídos. Esse processo é chamado de reconstrução densa,

pois tem como objetivo achar o valor em coordenadas de mundo do maior número de pontos da imagem possível, ao invés das poucas *features*. Esta etapa faz uso de todo o conhecimento da geometria epipolar adquirido no processo, para tornar o passo mais preciso e rápido.

Primeiro, uma série de transformações é aplicada às imagens de forma que os pontos correspondentes estejam na mesma coordenada-y em ambos os quadros, como mostra a Figura 14. Este passo é chamado de retificação da imagem e será melhor detalhado no próximo capítulo, sendo o foco do presente trabalho de graduação.

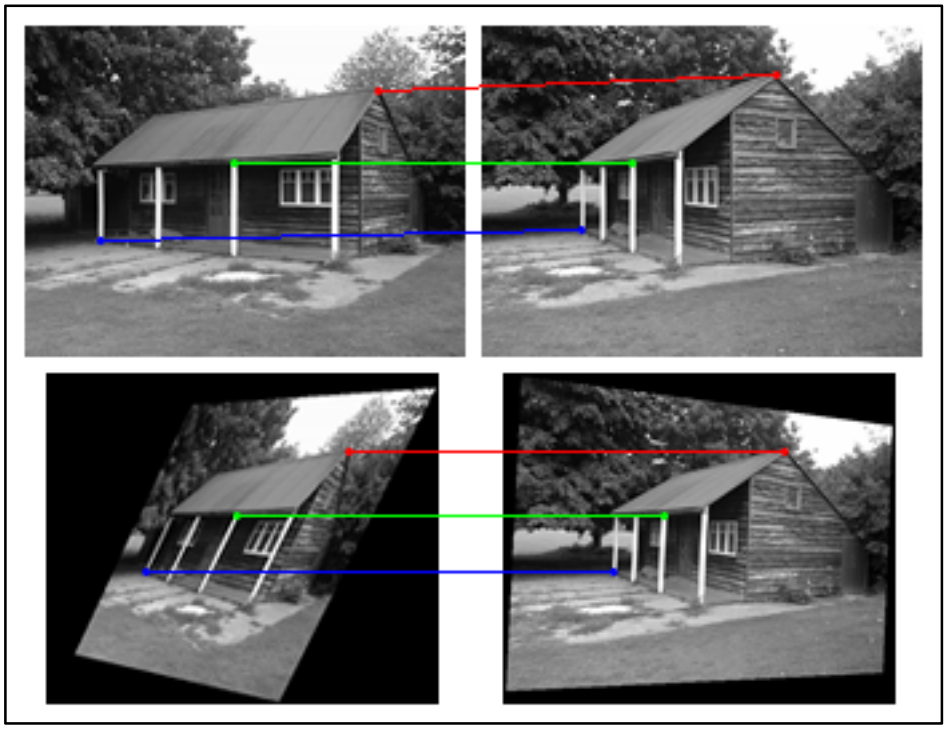

Figura 14: Acima, as imagens originais. Abaixo, as mesmas imagens após a retificação, com os pontos correspondentes na mesma coordenada-y

Com as imagens retificadas, a mesma técnica usada para selecionar as *features* no começo do processo pode ser aplicada para encontrar as correspondências para quase todos os *pixels*. Isto só é possível porque a busca será bem mais simples, já que ela se dará apenas por *pixels* da segunda imagem, localizados na mesma coordenada-y do ponto original na primeira figura. Além disto, algumas outras restrições podem ser adicionadas para

melhorar a performance da procura. Estas restrições dizem que, por exemplo, pontos na linha epipolar aparecem na mesma ordem em ambas as imagens, e que cada *pixel* tem um único correspondente na outra imagem.

Após as correspondências serem encontradas, uma nova matriz fundamental é estimada usando o algoritmo de *eight-point*. Ela será composta com as matrizes de parâmetros intrínsecos, formando a matriz essencial, para calcular o valor 3D desses pontos, utilizando as mesmas relações das reconstruções métricas e projetivas. Neste passo, a fundamental tende a ser mais precisa do que a encontrada durante o rastreamento das *features*, já que agora existe um aumento significativo no número de correspondências. O resultado da reconstrução densa pode ser visto na Figura 15.

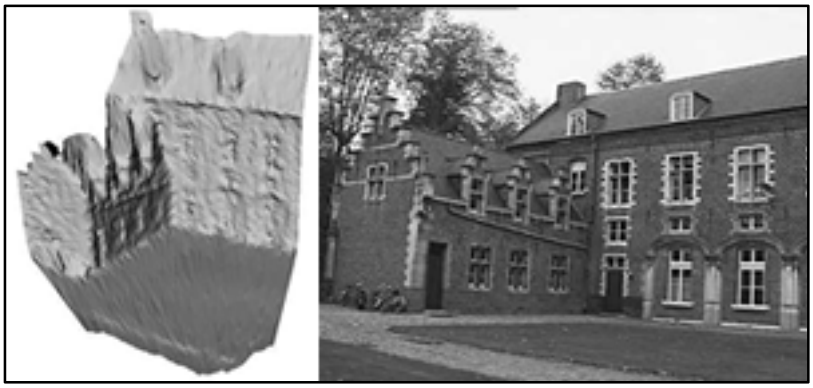

Figura 15: resultado da etapa de reconstrução densa [5]

3.7. Geração de malha e texturização

Com a posição espacial de praticamente todos os pontos da sequência inteira de imagens, pode-se juntar três deles para formar um plano. Juntando esses planos é possível gerar uma malha 3D de boa qualidade que represente de forma precisa e detalhada o objeto capturado.

Existem algumas formas de triangularizar uma nuvem de pontos. A mais simples é gerar o envoltório convexo de todos os pontos 3D, ou seja, o sólido de menor área lateral possível que contenha todos os pontos [17]. Entretanto, ela pode resultar numa forma bem diferente do objeto capturado. Uma alternativa que implica num modelo mais próximo do real consiste em

identificar locais onde a densidade de pontos é maior para tentar fazer a triangulação apenas entre esses mais próximos [18].

Depois de triangularizar os pontos, uma textura baseada na original é atribuída ao modelo 3D para que este fique ainda mais próximo do real. Um modo de texturização consiste em encontrar o centróide de cada triângulo da malha criada e, como a matriz de projeção é conhecida, projetar este ponto na imagem e verificar as informações de cor que este ponto vai possuir quando for mapeado na imagem. Este passo é feito para todas as imagens onde o centróide aparece e o valor de cor atribuído ao seu plano é a média aritmética da cor de todos os *pixels* que correspondem àquele centróide nos vários quadros. A Figura 16 mostra o resultado da triangulação e texturização de uma nuvem de pontos gerada a partir da reconstrução densa. Ela também é o resultado final de todo o processo de reconstrução.

Figura 16: resultado da triangulação e texturização da reconstrução densa obtida na **Figura 15** [5]

4. RETIFICAÇÃO DE IMAGENS

A retificação de imagens é um passo muito importante no processo de reconstrução densa. Praticamente todos os algoritmos para esta tarefa já partem do suposto que as imagens estão retificadas [19]. Mesmo a retificação de imagens sendo um problema clássico da área de visão computacional, existem poucos métodos para executar esta tarefa. De fato, até onde o autor conhece, existem apenas dois: a retificação planar e a retificação cilíndrica. Estas técnicas foram estudadas neste trabalho de graduação e uma implementação comparativa das mesmas foi realizada. Na sequência, o seu funcionamento é descrito, bem como os principais detalhes relacionados com a implementação.

Num processo de reconstrução, retificar as imagens significa aplicar transformações às elas de forma que as figuras obtidas possuam projeções epipolares correspondentes [16]. Em outras palavras, essas imagens são manipuladas de forma que linhas epipolares correspondentes se tornem colineares e paralelas ao eixo-x da imagem. Desta forma, o espaço de busca é reduzido à apenas uma dimensão. As vantagens de executar esta técnica vão além de simplesmente tornar a área de busca numa linha horizontal.

Ao fazer a retificação baseada na geometria epipolar das imagens, também é possível garantir outras restrições, acelerando e tornando mais consistente a procura por correspondências. A primeira restrição garante que todos os pontos de uma linha epipolar aparecerão na mesma ordem nas imagens retificadas, desde que esses objetos sejam opacos. Uma outra diz que cada ponto possui um único correspondente nas imagens seguintes. A Figura 17 ilustra isto. Nela, vê-se a representação de uma imagem, com l_1 sendo uma linha epipolar que cruza a foto e os *pixels* coloridos determinam os pontos da imagem que formam a linha. Ainda na Figura 17 pode-se ver a reprodução da mesma imagem, porém retificada; nela os *pixels* da linha epipolar aparecem uma única vez e na mesma ordem da imagem original.

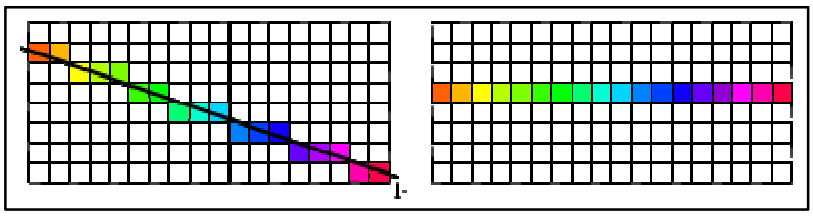

Figura 17: à esquerda, a representação de uma imagem; à direita, a representação de uma imagem retificada

Como já foi dito, duas formas de retificação são conhecidas e ambos determinam que o par de imagens a serem retificadas devam ser reorganizadas a partir de uma re-projeção. Os algoritmos diferem basicamente na forma como as figuras serão re-projetadas, como será mostrado a seguir. Os métodos retificam um par de imagens e o processo para retificar toda uma sequência de imagens é incremental. Na sequência, os métodos de retificação planar e cilíndrica serão apresentados.

4.1. RETIFICAÇÃO PLANAR

Este método de retificação é relativamente simples, tendo como ideia básica re-projetar todas as imagens num plano em comum paralelo à *baseline*. Assim, quando a figura for mapeada numa região em comum deste plano, têm-se a garantia que linhas epipolares correspondentes estarão na mesma altura.

Existem infinitos planos paralelos à *baseline* que poderiam ser usados para a retificação. Por simplicidade de cálculos, é escolhido um a partir da rotação dos planos de imagem, de forma que eles se tornem co-planares. Esta operação resulta numa nova matriz de projeção, baseada na original, e a garantia de que epipolos estão no infinito e, consequentemente, linhas epipolares são paralelas, como visto na Figura 18.

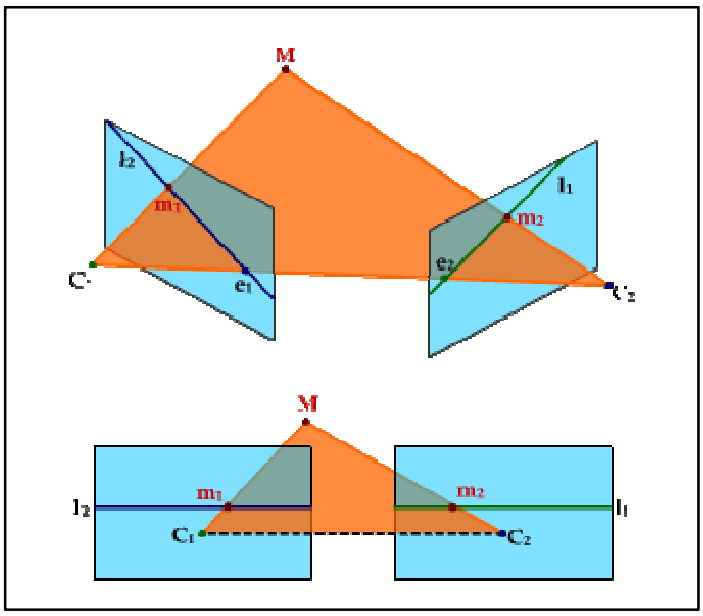

Figura 18: acima, um par de imagens. Abaixo, o mesmo par, porém retificado, com planos de imagem co-planares, epipolos no infinito e linhas epipolares paralelas

É necessário o conhecimento dos parâmetros de ambas as câmeras para que as novas sejam geradas. Elas vão diferenciar das antigas apenas na matriz de rotação e nos parâmetros intrínsecos, que serão os mesmos nas duas matrizes de projeção retificadas. Por outro lado, os centros de câmera serão os mesmos das câmeras originais. Desta forma, se as matrizes de projeção das duas imagens são:

$$P_1 = K_1[R_1 \mid -t_1] \quad \text{e} \quad P_2 = K_2[R_2 \mid -t_2], \tag{22}$$

as novas matrizes de projeção serão:

$$P_{R1} = K_R[R_R \mid -t_1] \quad \text{e} \quad P_{R2} = K_R[R_R \mid -t_2], \tag{23}$$

onde K_R será a nova matriz de parâmetros intrínsecos de ambas as câmeras, escolhida baseada nas matrizes de parâmetros intrínsecos K_1 e K_2 das imagens originais e R_R é a matriz que determina como os novos planos de imagem vão estar rotacionados. Ela é escrita da seguinte forma:

$$R_R = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix}, \qquad (24)$$

onde r_i são os eixos que formam o novo sistema de coordenadas da câmera. Como as imagens retificadas devem ser paralelas à *baseline*, estes eixos podem ser facilmente encontrados com o conhecimento das matrizes de projeção das imagens originais. Desta feita, r_1 será o novo eixo-x e este deve ser paralelo à *baseline*; r_2, o novo eixo-y, que deve ser ortogonal à r_1; e por fim, o novo eixo-z será r_3, que é ortogonal à r_1 e r_2. Assim, com z sendo o eixo-z de R_1:

$$r_1 = \frac{(t_1 - t_2)}{\|t_1 - t_2\|} \quad , \quad r_2 = z \times r_1 \quad \text{e} \quad r_3 = r_1 \times r_2 . \qquad (25)$$

A matriz K_R pode possuir qualquer valor. Em geral escolhe-se um baseado nas matrizes K_1 e K_2 das câmeras originais. Em [20] é recomendado usar a média das duas matrizes, assim:

$$K_R = \frac{(K_1 + K_2)}{2} . \qquad (26)$$

De posse da matriz de projeção retificada, é preciso calcular a matriz de transformação que vai levar a imagem obtida com P_1 para a retificada, obtida com P_{R1}. Esta matriz vai precisar desfazer a rotação aplicada à câmera original e, em seguida, rotacioná-la de acordo com a nova matriz de rotação calculada. Assim, como:

$$P_1 = [Q_1 | q_1] \ , \ P_2 = [Q_2 | q_2] \ , \ P_{R1} = [Q_{R1} | q_{R1}] \ \text{e} \ P_{R2} = [Q_{R2} | q_{R2}] \qquad (27)$$

as matrizes de transformação que retificarão as imagens serão:

$$T_1 = Q_{R1} \cdot Q_1^{-1} \quad \text{e} \quad T_2 = Q_{R2} \cdot Q_2^{-1} . \qquad (28)$$

Depois que a transformação T_i for aplicada à todos os *pixels* da imagem original, ela estará retificada.

Este método de retificação de imagens é bem simples, além de possuir uma execução rápida, pois depende apenas do cálculo de uma matriz de transformação para cada imagem. Porém, ele não é robusto o suficiente para qualquer tipo de movimento de câmera. Para casos onde o deslocamento de uma pose de câmera em relação à outra foi muito grande, o processo de retificação pode falhar.

Nas situações onde o movimento da câmera foi muito brusco, os epipolos tendem a se aproximar da imagem, gerando, graças a isto, imagens cada vez maiores. Desta feita, o resultado da retificação será um par de imagens bastante distorcido, tornando impreciso o processo de reconstrução densa.

Esta técnica falha quando os epipolos se aproximam tanto da imagem que ficam dentro dela. Para esses casos, que ocorrem quando a câmera dá um *zoom* na imagem, por exemplo, a imagem retificada terá tamanho infinito, pois os pontos que estão na *baseline* são paralelos ao plano em comum, tendo sua intersecção com este apenas no infinito, como mostra a Figura 19.

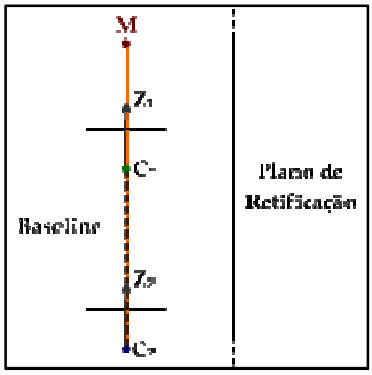

Figura 19: caso onde a câmera se deslocou para frente, fazendo com que a projeção do ponto M, na *baseline*, nunca intercepte o plano de projeção

4.2. RETIFICAÇÃO CILÍNDRICA

Para ambientes controlados, onde a câmera terá movimento constante, suave e nenhum *zoom* será aplicado, a retificação planar pode ser utilizada, pois é bastante simples e muito eficiente nestas situações. Porém, nem sempre é

possível ter estas condições. Às vezes, a captura é realizada à mão livre, causando imperfeições como movimentos para frente, caracterizando um caso onde o epipolo está dentro da imagem.

Para tratar esses casos, [21] propôs uma nova técnica mais robusta e geral para retificar um par de imagens. Ela se diferencia da retificação planar, principalmente, por, ao invés de usar um plano em comum, usar um cilindro em comum para re-projetar o par de imagens.

O método consiste em determinar um cilindro de raio unitário que tem a *baseline* com eixo de revolução e, em seguida, mapear cada *pixel* da imagem numa coordenada (z, θ) de um sistema de coordenadas cilíndrico. Desta feita, cada ponto (z, θ) da imagem retificada é usado normalmente, como se fosse um ponto (u, v) na imagem.

Para realizar este mapeamento cada linha epipolar deve ser rotacionada livremente no espaço até que esteja paralela à *baseline*. Em seguida, estas linhas são projetadas no cilindro. Desta forma, como linhas epipolares correspondentes serão rotacionadas em um mesmo ângulo, ao final do processo elas possuirão a mesma coordenada-θ na imagem final. Desta feita, cada ponto m na imagem é transformado num ponto m_R na imagem retificada, a partir da seguinte relação:

$$m_R = \underbrace{S_C \cdot T_C \cdot R_C}_{L} \cdot m, \qquad (29)$$

onde m deve estar em coordenadas de câmera, R é a matriz que vai rotacionar a linha epipolar na qual o ponto m está contido até que ela fique paralela à *baseline*, T é uma matriz de mudança de base que levará m de coordenadas de câmera para coordenadas cilíndricas e S é a matriz que projetará m no cilindro de raio unitário.

Caso se tenha o ponto m_R, o ponto m pode ser encontrado facilmente da seguinte forma:

$$m = \underbrace{R_C^T \cdot T_C^T \cdot S_C^{-1}}_{L^{-1}} \cdot m_R . \qquad (30)$$

O sistema de coordenadas cilíndricas é formado por três vetores, X_C, Y_C, Z_C, e um ponto de origem O_C, que será igual à origem do sistema de coordenadas da primeira câmera. Assim, a matriz T_1 de mudança de base da primeira imagem será escrita como:

$$T_{C1} = \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}, \qquad (31)$$

onde X_C é um vetor que terá o mesmo sentido da *baseline*, Y_C será o produto vetorial do vetor $z = (0, 0, 1)^T$ com X_C, e Z_C é um vetor ortogonal à X_C e Y_C. Dito isto, eles serão descritos por:

$$X_C = R_1^T \cdot (t_2 - t_1) \quad , \quad Y_C = z \times X_C \quad \text{e} \quad Z_C = X_C \times Y_C . \qquad (32)$$

Como o cilindro de projeção é comum para ambas as imagens, os vetores que formam a base para a matriz de transformação T_2 são definidos em relação aos vetores da base de T_1. Assim, usando as matrizes de rotação de ambas as câmeras, é possível encontrar T_2 de forma mais simples:

$$T_{C2} = T_{C1} \cdot R_1^T \cdot R_2 . \qquad (33)$$

Para rotacionar uma linha epipolar contendo um ponto m, em coordenadas de câmera, até que ela seja paralela à *baseline* é necessário encontrar a matriz de rotação R_C para cada imagem. Ela pode ser calculada usando dois vetores, o z′, que é a projeção do vetor normal do plano de imagem no plano epipolar, e o vetor p′, que é a projeção do ponto m no plano normal do cilindro. Eles são encontrados da seguinte forma:

$$z' = axis \times (z \times axis) \quad \text{e} \quad p' = T_C^T \cdot B \cdot T_C \cdot m , \qquad (34)$$

onde

$$\text{axis} = (t_2 - t_1) \times m \quad \text{e} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{35}$$

Com os dois vetores, R_C é encontrado como:

$$R_C = \begin{bmatrix} p' \\ p' \times z' \\ (p' \times z') \times p' \end{bmatrix}^T \cdot \begin{bmatrix} z' \\ p' \times z' \\ (p' \times z') \times z' \end{bmatrix}. \tag{36}$$

O último passo é encontrar a matriz S_C de cada imagem do par, que projeta a linha epipolar, agora em coordenadas cilíndricas, num cilindro de raio unitário, ou seja:

$$\left\| B \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{k} & 0 \\ 0 & 0 & \frac{1}{k} \end{bmatrix} \cdot T_C \cdot R_C \cdot m \right\| = 1, \tag{37}$$

o que leva à conclusão de que $k = \left\| B \cdot T_C \cdot R_C \cdot m \right\|$.

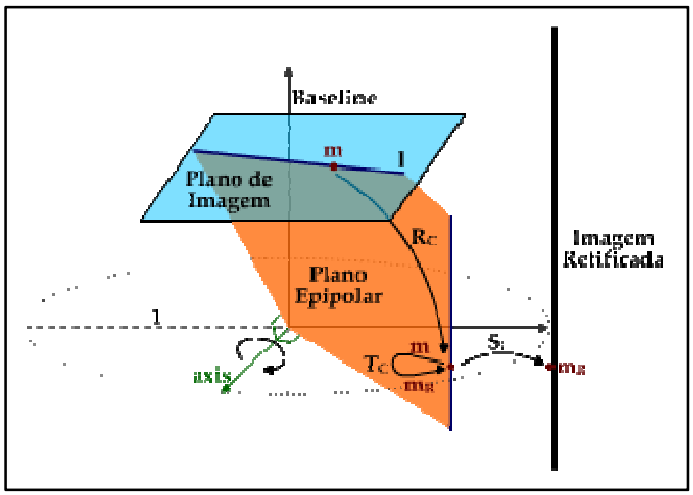

Figura 20: sequência de atividades da retificação cilíndrica

Para os casos onde o epipolo está dentro da imagem, é necessário um cuidado maior na escolha dos vetores X_C, Y_C e Z_C, afetando a sequência inteira de execução. Além disso, todos esses cálculos relativamente complexos, que são realizados num espaço tridimensional, devem ser feitos para cada *pixel* da imagem. Por isso, apesar de tratar todos os movimentos de

câmera, a retificação cilíndrica exige um tempo de execução altíssimo, já que toda linha epipolar deve ser rotacionada e projetada, como mostra a Figura 20.

4.3. RETIFICAÇÃO CILÍNDRICA SIMPLIFICADA

A retificação cilíndrica, porém, pode ser bastante simplificada se as informações da geometria projetiva e epipolar forem usadas, fazendo com que todos os cálculos sejam feitos no plano de imagem, e evitando assim operações tridimensionais.

A ideia deste método é a mesma do anterior: reparametrizar a imagem num sistema de coordenadas cilíndrico. Entretanto, ele difere no cilindro escolhido. Enquanto na implementação convencional o cilindro é centrado na *baseline*, na forma simplificada as transformações ocorrem ao redor dos epipolos e, como estes estão no mesmo plano da imagem, nenhuma operação ocorrerá no espaço tridimensional [22].

Como pode ser visto na Figura 21, cada linha epipolar possui um ângulo θ em relação ao epipolo, assim como cada *pixel* dela está a uma distância r deste mesmo ponto. Desta forma, as linhas epipolares são reescritas horizontalmente na nova imagem. No final, o par estará retificado porque linhas epipolares correspondentes possuem o mesmo ângulo em relação ao epipolo, já que elas estão no mesmo plano epipolar.

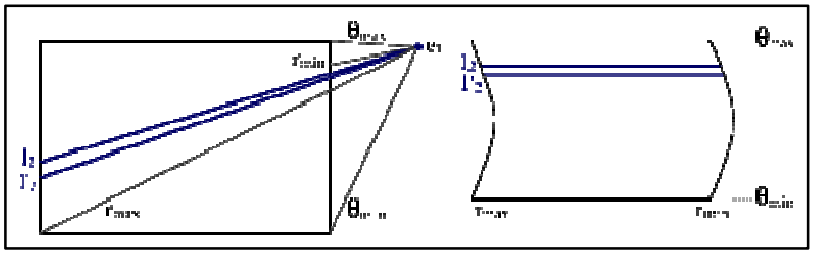

Figura 21: representação da retificação cilíndrica simplificada para uma imagem

A primeira etapa a ser realizada neste tipo de retificação é determinar as regiões comuns à ambas imagens, que são as partes que aparecem tanto numa figura, quanto na outra. Para isso, é necessário encontrar as linhas

epipolares extremas de cada imagem, aquelas que irão possuir o maior e o menor ângulo. Essas linhas sempre passarão por algum vértice da imagem e podem ser facilmente encontradas conhecendo a informação sobre que região pertence o epipolo.

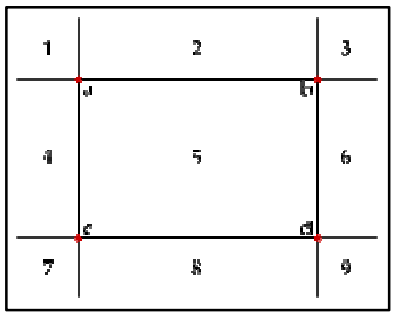

Figura 22: regiões onde o epipolo pode estar localizado

Na Figura 22 pode ser visto que existem nove regiões onde o epipolo pode ser encontrado. Em cada uma delas, as linhas epipolares extremas vão cruzar um dos quatro vértices. Por exemplo, caso o epipolo esteja localizado na região 3, suas linhas epipolares mais externas serão aquelas que passarão pelo epipolo e os cantos a e d. Caso o epipolo esteja na região 5, consequentemente dentro da imagem, podem ser escolhidas as linhas que vão do epipolo para qualquer um dos pares de vértices opostos entre si, ou seja, a e d ou b e c. Este processo é usado para encontrar as linhas epipolares extremas em ambas as imagens.

De posse dessas linhas epipolares mais externas, é encontrada a região em comum a ambas imagens. Para isso é necessário encontrar as linhas epipolares correspondentes às linhas extremas para ambas as imagens, como mostra a Figura 23, onde as linhas epipolares l'_{1max} e l'_{1min} na segunda imagem, por exemplo, são as correspondentes das linhas epipolares extremas l_{2max} e l_{2min} da primeira imagem.

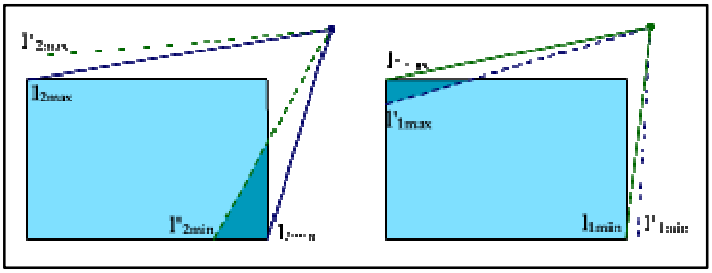

Figura 23: linhas epipolares extremas e região em comum

As linhas correspondentes são calculadas a partir da matriz de homografia H definida a partir da geometria epipolar. Assim, elas são encontradas da seguinte maneira:

$$l'_{1max} = H^{-T} \cdot l_{2max},\ l'_{1min} = H^{-T} \cdot l_{2min},\ l'_{2max} = H^{T} \cdot l_{1max} \text{ e } l'_{2min} = H^{T} \cdot l_{1min}. \quad (38)$$

Com isso, as duas linhas extremas são aquelas que interceptam a imagem em mais de uma aresta. No caso da **Figura 23**, as linhas escolhidas seriam l'_{2min} ou l_{1min}, já que elas são correspondentes, e l_{2max} ou l'_{1max}. Em geral, a primeira imagem é usada como referência.

Após conhecer as linhas extremas, as imagens retificadas já podem ser construídas. Primeiro, a linha externa mínima, aquela que possui o menor ângulo θ, é reescrita horizontalmente na nova imagem. Em seguida, a linha correspondente a esta, na segunda figura, é encontrada usando a matriz de homografia para também ser reescrita horizontalmente na imagem retificada.

Posteriormente, esta linha epipolar extrema é rotacionada em $\Delta\theta$ graus ao redor do epipolo para que uma nova linha epipolar, que passe por outros pontos da imagem, seja encontrada. Este ângulo de rotação não pode ser grande, pois poderiam ocorrer situações em que a linha epipolar seguinte estaria tão distante que alguns *pixels* não seriam reescritos na imagem retificada, já que nenhuma linha epipolar passaria por eles, causando assim uma perda de informação da imagem. Para evitar isto, é escolhido o menor ângulo possível para esta rotação que, no pior caso, será o deslocamento de um *pixel* na borda oposta ao epipolo, como mostrado na Figura 24. Desta

forma, têm-se a garantia que todos os *pixels* da imagem serão, pelo menos, preservados na retificação.

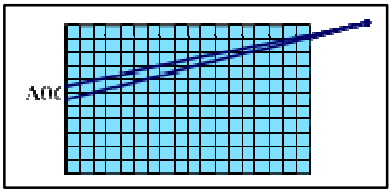

Figura 24: menor ângulo entre duas linhas consecutivas

Esta nova linha encontrada será reescrita horizontalmente na imagem retificada. Em seguida, a linha correspondente a ela na outra foto é encontrada para, também, ser reescrita em sua respectiva imagem retificada.

Este processo de rotacionar a linha epipolar ao redor do epipolo em um ângulo $\Delta\theta$, reescrevê-la na imagem retificada, achar o seu correspondente na foto seguinte para também reparametrizá-la é repetido, linha a linha, até que todas as linhas epipolares até a linha extrema máxima sejam reescritas. Cada uma dessas linhas corresponde a um ângulo ao redor do epipolo, do mínimo ao máximo, na imagem retificada.

O algoritmo de Bresenham [23], utilizado para desenhar as linhas, determina que *pixels* da imagem serão extraídos para cada linha epipolar. Como esta técnica não suaviza as linhas, pode-se dizer que o tamanho da linha na imagem retificada permanecerá igual ao da linha na imagem original, assim, a figura resultado da retificação terá uma largura, no máximo, igual ao maior tamanho entre a largura e a altura da imagem original. A altura da retificação será , no máximo, duas vezes a soma da altura com a largura da imagem original.

Nas situações onde o epipolo está dentro da imagem, não é necessário realizar nenhuma mudança no algoritmo, pois ele já trata esta caso, o que torna ainda mais fácil a implementação desta técnica. Além disto, como a matriz de homografia é a mesma para todas as linhas epipolares, ela é calculada apenas uma vez. Finalmente, todas as operações são feitas no plano

de imagem, de forma que o tempo de execução também é bem mais baixo se comparado com a retificação cilíndrica tradicional.

4.4. IMPLEMENTAÇÃO

Com base no estudo realizado foi implementada a técnica de retificação cilíndrica simplificada e uma versão parcial da retificação planar disponível em [15] foi concluída pelo autor.

A implementação da retificação cilíndrica foi feita usando o Matlab [24], versão R2007b. Nela, nenhuma biblioteca específica deste *software* foi utilizada, com exceção daquelas destinadas a carregar e salvar imagens. A escolha pelo desenvolvimento em Matlab ocorreu graças à simplicidade com que esta ferramenta trata os problemas matemáticos, possibilitando uma escrita mais rápida de *softwares* com este tipo de conteúdo em comparação com as linguagens de programação convencionais, como C++. Originalmente, a proposta deste trabalho previa a utilização desta linguagem, porém optou-se por implementar a primeira versão da técnica em Matlab, principalmente, porque um tempo considerável foi empregado no estudo e análise comparativa das técnicas de retificação para escolher a mais adequada ao problema da reconstrução densa e na posterior implementação da técnica selecionada. Desta feita, o uso do Matlab garantiria a conclusão da retificação cilíndrica em sua totalidade. Neste momento, o desenvolvimento de uma segunda versão da técnica em C++ está em andamento, no sentido de fornecer uma implementação que poderá ser facilmente integrada ao *pipeline* de reconstrução 3D.

Para que a retificação cilíndrica seja realizada é necessário, além do par de imagens, o conhecimento da matriz fundamental. No estágio do processo de reconstrução em que o *dense matching* está localizado, esta matriz já é conhecida. Entretanto, como apenas a retificação foi implementada neste trabalho, um *script* auxiliar foi adaptado de [8] e incorporado ao nosso *software* com o objetivo de calcular a fundamental e garantir a completude da solução.

Uma implementação parcial da retificação planar descrita no capítulo 4, disponibilizada em [15], foi concluída e usada para comparação de resultados com a retificação cilíndrica. Devido à complexidade e falta de informações mais detalhadas, a retificação cilíndrica tradicional não foi implementada, sendo desenvolvida apenas a sua forma simplificada.

5. RESULTADOS

Para comprovar a eficácia da técnica de retificação cilíndrica simplificada, foram realizados alguns estudos de caso. Além disso, foi realizada uma comparação da retificação cilíndrica simplificada com a retificação planar, de maneira que as vantagens e desvantagens de cada uma também pudessem ser avaliadas.

Quatro casos foram analisados como forma de avaliar a retificação cilíndrica simplificada: "*move house*" e "*temple*", onde a captura das imagens foi feita de forma controlada; além de "agamenon" e "*desktop*", com a aquisição obtida sem a ajuda de nenhum equipamento especial. Além disto, os resultados também foram comparados com os obtidos pela retificação planar dos mesmos pares de imagens. Três critérios foram usados neste confronto. Um deles foi o de precisão na igualdade da coordenada-y nas duas imagens. Para isto, foram escolhidos aleatoriamente 36 *pixels* numa das imagens retificadas pela técnica planar e, em seguida, as suas coordenadas-y foram comparadas com as de seus correspondentes na outra imagem. Os mesmos 36 *pixels* foram escolhidos numa das imagens retificadas pelo método cilíndrico simplificado e seu *pixel* correspondente também foi analisado. Os outros dois critérios foram o tamanho da imagem retificada gerada pelas técnicas e o tempo médio necessário para as execuções de cada técnica, sem incluir o cálculo da matriz fundamental quando necessário.

Todos os casos foram executados dez vezes, com seu tempo medido em cada uma delas, num computador possuindo um processador Athlon 3200+ com velocidade de 4,00 GHz e 1GB de memória RAM.

É importante mencionar que, como a qualidade das imagens pode ficar bastante comprometida durante a impressão, a ponto de comprometer a visualização dos resultados, as imagens que ilustram os estudos de caso a partir deste ponto do capítulo podem ser encontradas em sua resolução original no seguinte endereço eletrônico: http://www.cin.ufpe.br/~rar3/tg.

5.1. MOVE HOUSE

No primeiro caso foram usados dois quadros distintos de um vídeo, ambos possuindo 512 *pixels* tanto de altura quanto de largura, como mostra a Figura 25. Nela, assim como em todas as imagens que ilustram os casos de testes, foram selecionados três pontos aleatoriamente e eles foram ligados a seus correspondentes na outra foto com o objetivo de mostrar a variação da coordenada-y de cada um deles. A filmagem foi realizada de forma controlada, onde a câmera estava parada e o que se movia era a maquete. Desta forma, garantiu-se que nenhum fator, como movimento brusco da câmera ou variação de iluminação no ambiente atrapalhasse a aquisição das imagens. Além disso, os parâmetros intrínsecos e extrínsecos da câmera eram conhecidos.

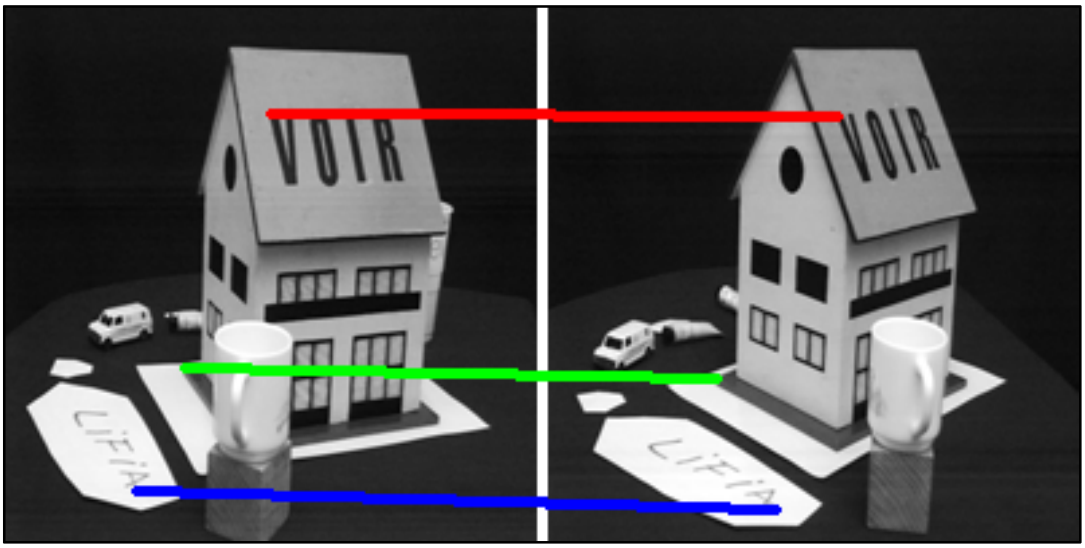

Figura 25: par original da sequência conhecida como "*move house*" [25]

O par de imagens foi retificado usando ambas as técnicas. A planar concluiu o processo em uma média de 2,2706 segundos e resultou numa imagem com 634 *pixels* de largura e 590 de altura. Já a retificação cilíndrica simplificada gerou uma imagem de 512 *pixels* de largura e 989 de altura em uma média de 18,2712 segundos. As Figura 26 e Figura 27 mostram os resultados da retificação planar e cilíndrica simplificada, respectivamente.

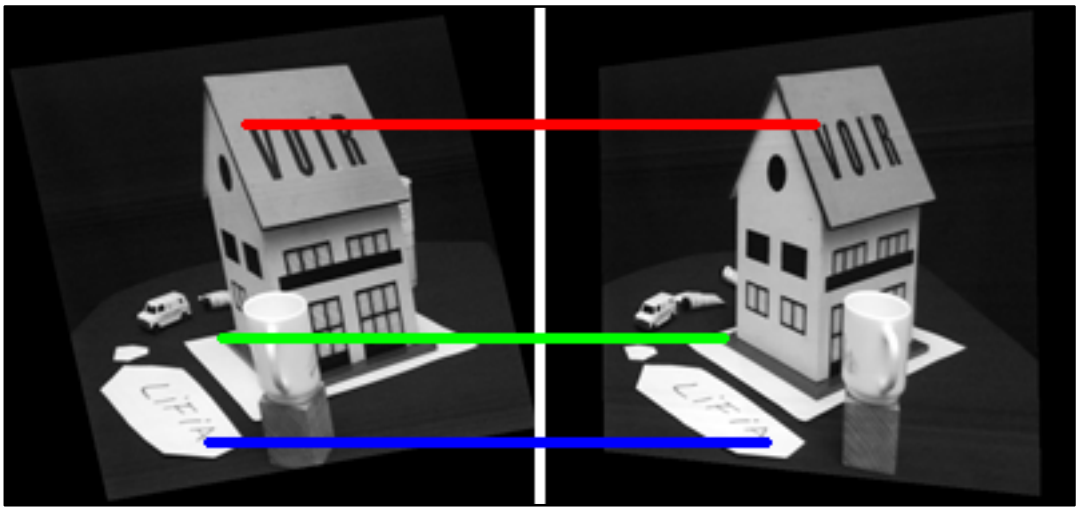

Figura 26: retificação planar da cena "*move house*"

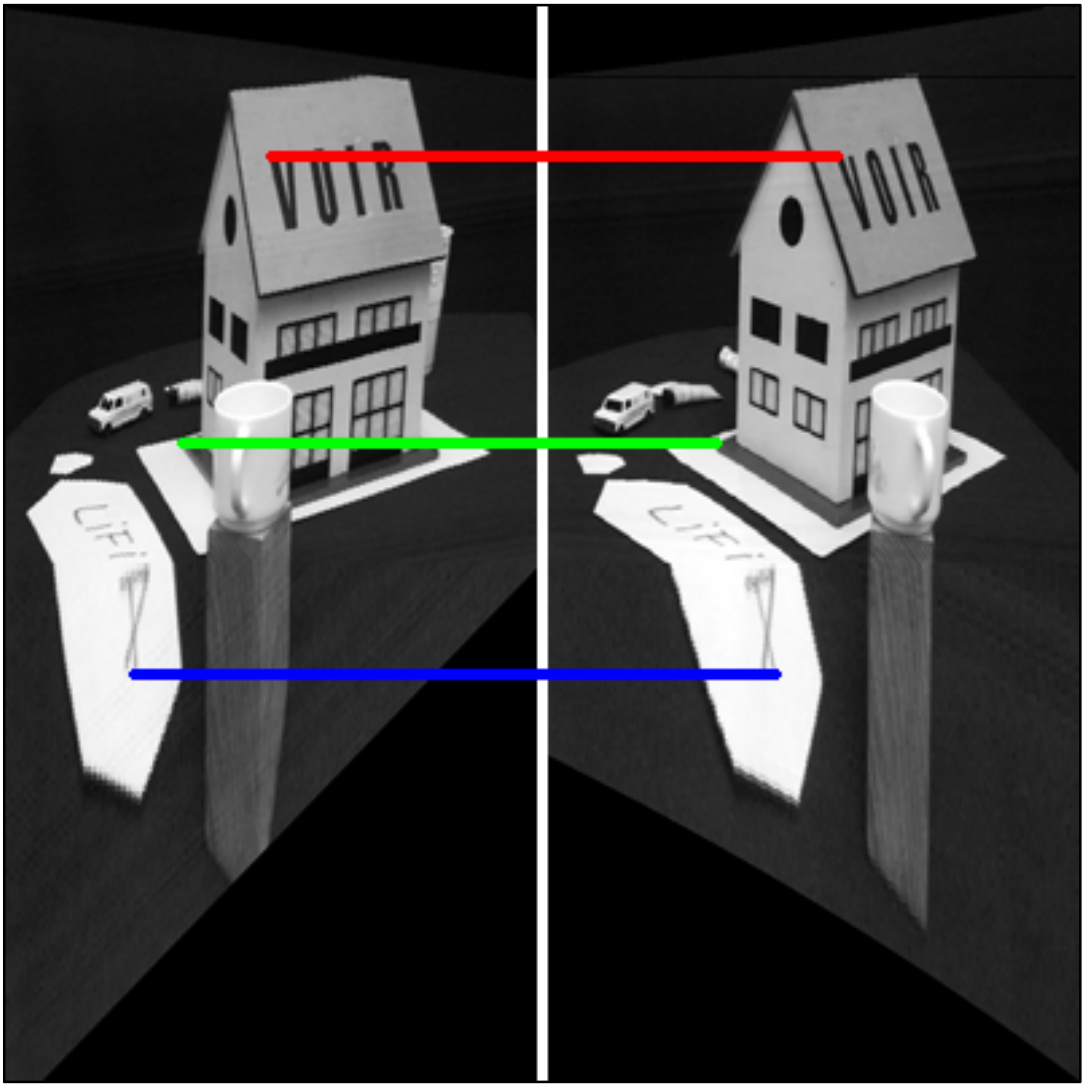

Figura 27: retificação cilíndrica simplificada da cena "*move house*"

Neste caso, onde a captura foi controlada, a retificação planar se mostrou mais vantajosa, resultando numa imagem aproximadamente 74% menor. Quanto maior a imagem, maior a distorção dos *pixels*, pois pode ser que dois ou mais deles sejam necessários para representar um único ponto da imagem original, causando imprecisão nos passos seguintes da reconstrução densa.

Ambas as técnicas obtiveram resultados satisfatórios em termos de precisão. Em ambas as formas de retificação, para nenhum ponto foi encontrada uma diferença maior que dois *pixels*, para cima ou para baixo, entre o escolhido e o seu correspondente na outra imagem.

5.2. TEMPLE

No segundo caso, as duas técnicas foram avaliadas utilizando um par de imagens extraídas de uma sequência de fotografias de uma maquete de um templo, onde cada imagem possui um tamanho de 480 por 640 *pixels*. As fotos também foram obtidas com uso de equipamentos especiais, como trilhos para garantir que nenhum movimento indesejado da câmera fosse realizado. A diferença deste par em relação ao do primeiro caso é que neste a câmera se moveu menos, fazendo com que o epipolo fique mais próximo da imagem. Este fato pode ser observado na Figura 28, onde os pontos correspondentes, representados pelas duas extremidades das linhas vermelha, verde e azul, praticamente já possuem a mesma altura.

O par de imagens também foi retificado por ambos os métodos, demorando em média 2,7567 segundos para ser executado pela retificação planar, o que resultou numa imagem com 783 *pixels* de largura e 626 de altura, como pode ser vista na Figura 29. Já a retificação cilíndrica simplificada deste mesmo par gerou uma imagem de 640 *pixels* de largura por 1029 de altura, resultado mostrado pela Figura 30, em 12,8328 segundos na média.

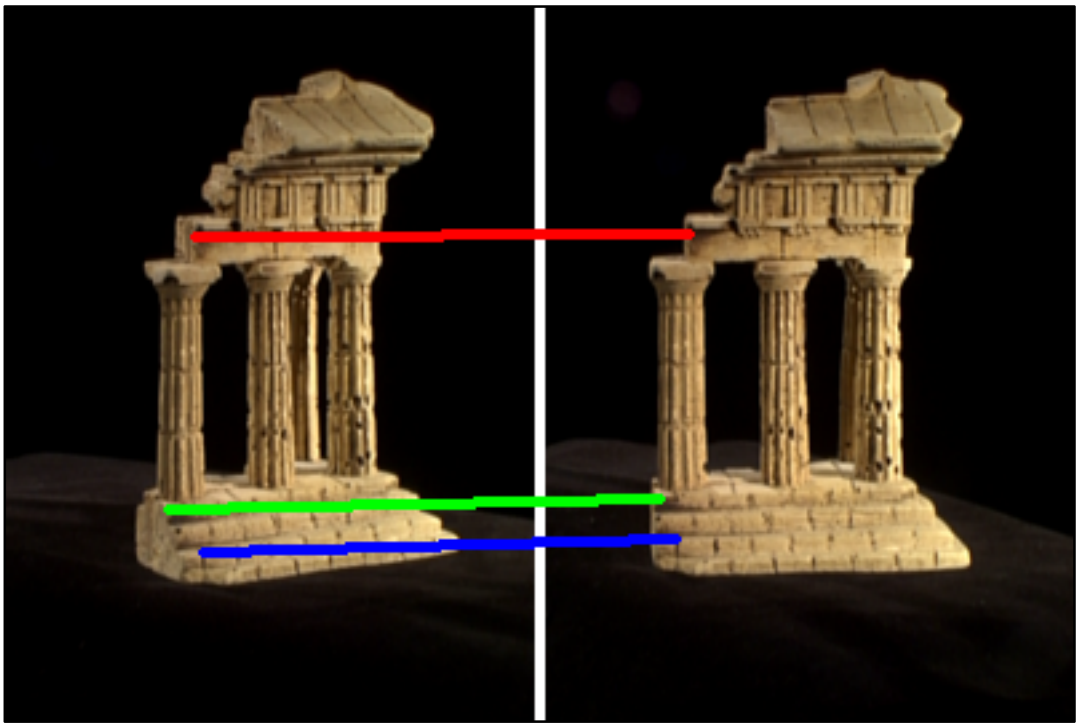

Figura 28: par original da sequência conhecida como "*temple*" [26]

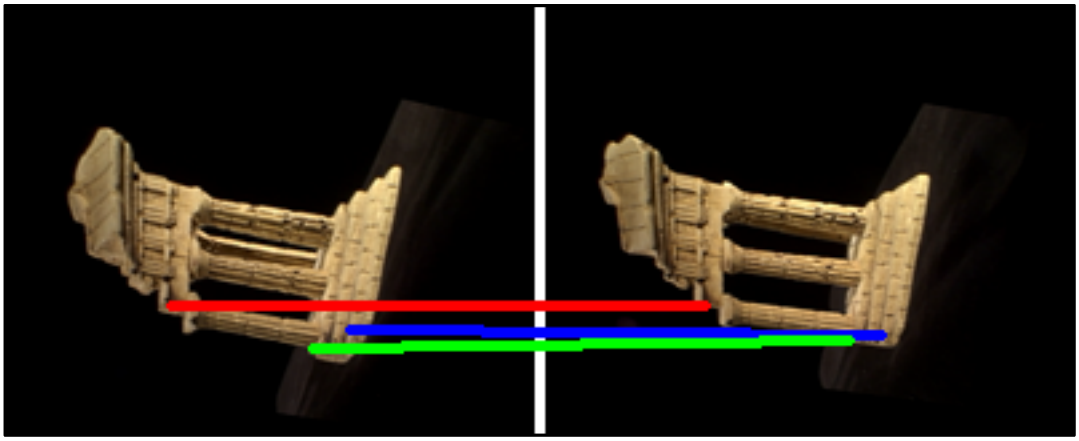

Figura 29: retificação planar da cena "*temple*"

Para este par a retificação planar começou a mostrar algumas deficiências. Mesmo tendo gerado uma 73% imagem menor, a precisão foi um problema. Dos 36 *pixels* escolhidos para verificar a precisão nestas fotos, 28 apresentaram uma diferença entre seus correspondentes maior que dois *pixels* na coordenada-y, contra 9 da retificação cilíndrica simplificada. Esta imprecisão fica visível se observarmos o ponto marcado pela linha verde na Figura 29. Para este ponto a diferença na coordenada-y foi maior na imagem retificada em comparação com a original.

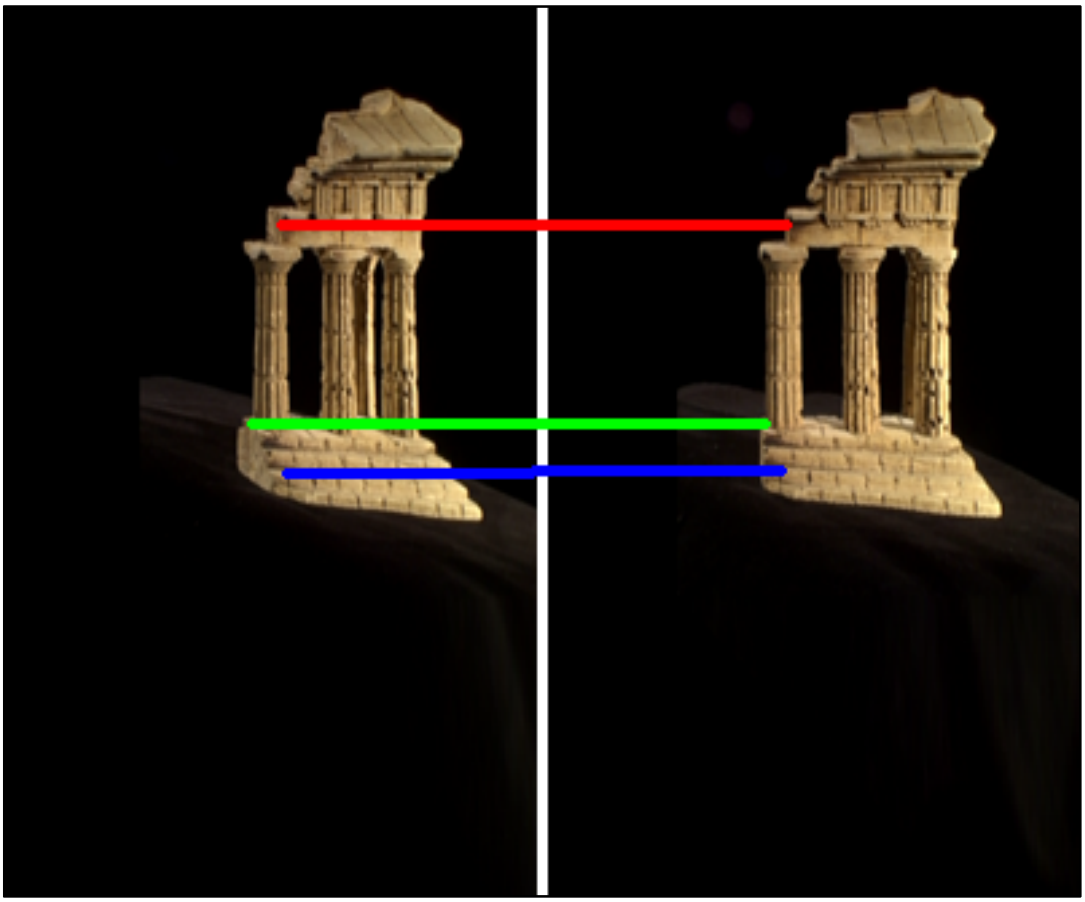

Figura 30: retificação cilíndrica simplificada da cena "*temple*"

5.3. AGAMENON

Em um terceiro caso, os métodos de retificação foram analisados com imagens capturadas livremente pelo autor, sem nenhum equipamento especial para a aquisição ou trato especial do cenário. Para isto, duas fotografias foram tiradas do alto de um edifício, com a câmera capturando num movimento livre. Essas imagens, com tamanho de 1024 *pixels* de largura por 768 de altura, podem ser vistas na Figura 31.

Para este caso, a forma planar retificou o par de imagens em 22,8313 segundos, em média, resultando numa tamanho de 1954 por 1485 *pixels*. Já a retificação cilíndrica simplificada gerou uma imagem 62% menor, com 1024 pontos de largura e 1766 de altura em 68,1882 segundos, na média. A Figura 32 e Figura 33 mostram esses resultados.

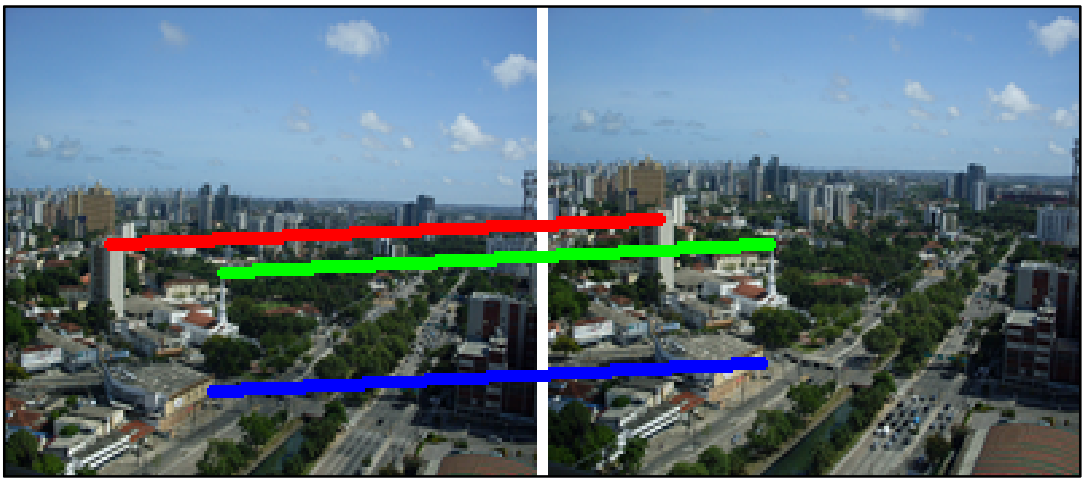

Figura 31: par original da sequência conhecida como "agamenon"

Assim como no primeiro caso de teste, não foi encontrada uma diferença maior que dois *pixels* no eixo-y entre os 36 pontos correspondentes selecionados em ambas as técnicas, o que mostra que a retificação planar não depende do fato da captura ser controlada para ser realizada corretamente. Porém, é necessário que a distância entre a posição das duas câmeras não seja muito pequena, para não ocorrerem os problemas do caso anterior, e nem muito grande, para não correr o risco do epipolo ficar dentro da imagem. Entretanto, este controle é muito difícil de ser obtido numa captura livre, o que torna a retificação planar uma técnica menos robusta e flexível.

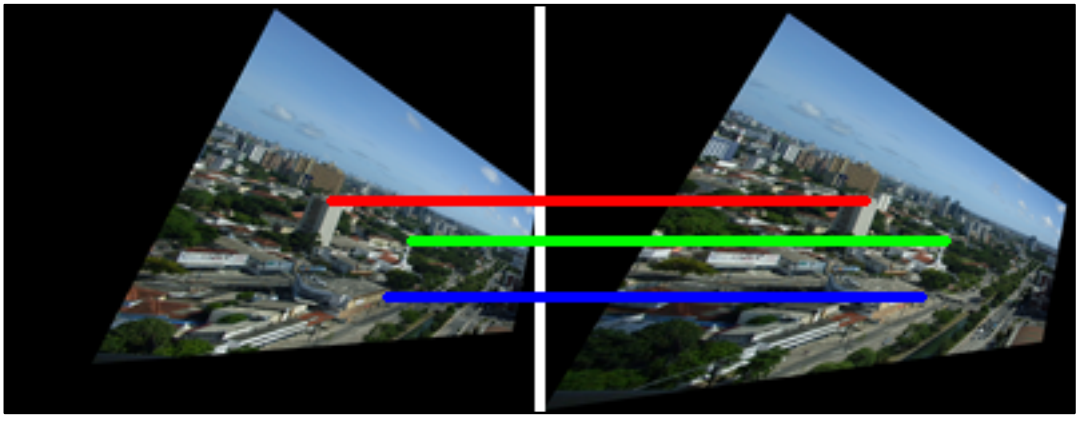

Figura 32: retificação planar da cena "agamenon"

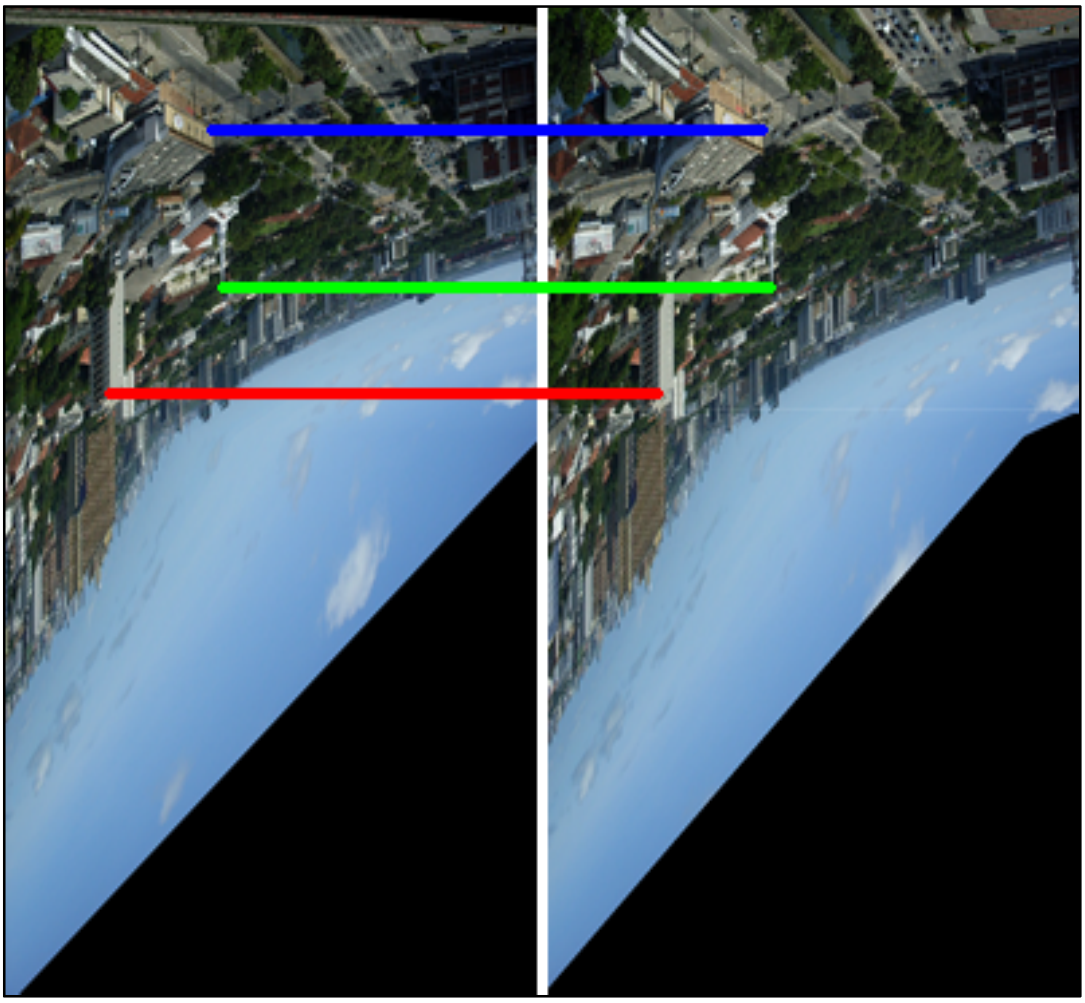

Figura 33: retificação cilíndrica simplificada da cena "agamenon"

5.4. DESKTOP

Por fim, no quarto caso foi observado como os métodos de retificação se comportam quando a diferença entre a posição da câmera que capturou as duas imagens se diferencia por um movimento para frente (*zoom*), caracterizando uma situação onde o epipolo está dentro da imagem. Para representar esta situação, foi tirada pelo autor uma foto de uma mesa e, em seguida, sem aplicar nenhum movimento de rotação ou translação na câmera, foi dado um *zoom*. Tecnicamente, o *zoom* nas câmeras digitais representa deslocar o plano de projeção, juntamente com o seu centro, para frente, caracterizando o movimento desejado. As imagens obtidas possuem 768 *pixels* de largura por 576 de altura e são mostradas na Figura 34.

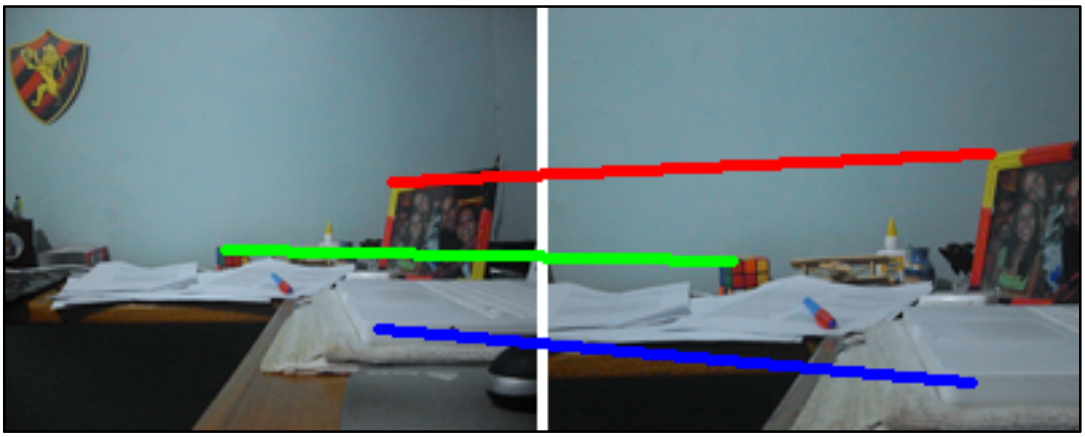

Figura 34: par original da sequencia conhecida como "*desktop*"

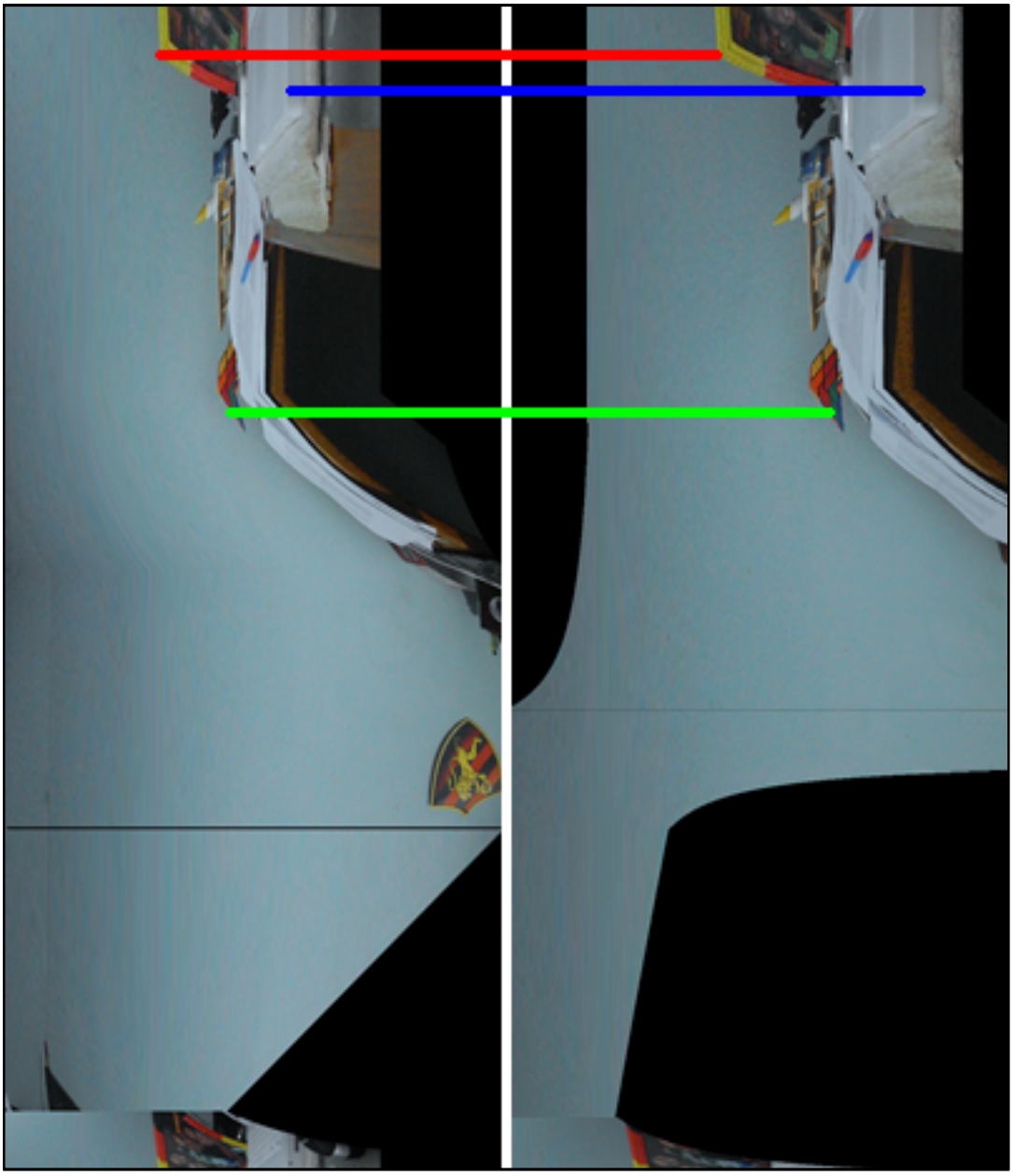

Figura 35: retificação cilíndrica simplificada da cena "*desktop*"

Como esperado, a técnica de retificação planar não conseguiu tratar este caso. Por outro lado, a retificação cilíndrica simplificada não encontrou problemas para retificar a imagem de forma satisfatória. A imagem foi gerada em 44,5835 segundos, em média, possuindo 768 *pixels* de largura e 1748 *pixels* de altura. Ela foi retificada com bastante precisão, com apenas um dos 36 pontos selecionados apresentando uma distância maior que dois *pixels* na coordenada-y do seu correspondente. O resultado é mostrado na Figura 35.

6. CONCLUSÕES

A retificação cilíndrica simplificada de imagens desenvolvida neste trabalho de graduação obteve bons resultados nos três quesitos avaliados. Se comparada com a retificação planar, como pode ser visto na Tabela 1, a abordagem tema deste trabalho foi superior no que tange a precisão, em todos os casos. Até nas situações onde a retificação cilíndrica simplificada teve algum problema de imprecisão, a diferença na coordenada-y de onde o ponto deveria estar para onde ele realmente ficou nunca foi maior do que 5 *pixels*, ao passo que a planar chegou a produzir uma diferença na altura de 13 *pixels*.

Em relação ao tamanho da imagem, em alguns casos a retificação planar obteve resultados melhores. Porém, as dimensões da imagem retificada neste método crescem à medida que o epipolo se aproxima da imagem, podendo gerar figuras muito grandes, ao passo que na retificação cilíndrica simplificada é garantido que o resultado retificado terá um tamanho fixo máximo.

		Tempo de execução	Tamanho original	Tamanho retificado	*Pixels* errados	Maior erro
Move House	**CS**	18,2712 s.	512 x 512	512 x 989	0	-
	P	2,2706 s.	512 x 512	634 x 590	0	-
Temple	**CS**	12,8328 s.	480 x 640	640 x 1029	9	5 *pixels*
	P	2,7567 s.	480 x 640	783 x 626	28	13 *pixels*
Agamenon	**CS**	68,1882 s.	1024 x 768	1024 x 1766	0	-
	P	22,8313 s.	1024 x 768	1954 x 1485	0	-
Desktop	**CS**	44,5835 s.	768 x 576	768 x 1748	1	5 *pixels*
	P	-	-	-	-	-

Tabela 1: comparativo dos resultados obtidos pela retificação cilíndrica simplificada (CS) e planar (P) para cada um dos casos avaliados

A principal conclusão extraída a partir do estudo dos métodos de retificação e da análise dos resultados obtidos nos experimentos é a de que a retificação cilíndrica simplificada realmente é mais robusta, já que consegue tratar todos os tipos de movimentos de câmera possíveis. Entretanto, como o processo é mais lento em comparação com a retificação planar, que em

algumas situações possui resultados igualmente bons, às vezes pode ser mais conveniente utilizar esta segunda abordagem.

Portanto, para *softwares* de reconstrução onde o principal foco é a robustez, é extremamente conveniente que a técnica escolhida para a retificação de imagens seja a cilíndrica. Mas, caso se deseje robustez com desempenho computacional, o ideal é o desenvolvimento de uma técnica híbrida que analise o movimento da câmera, já que esta informação já está disponível no estágio de reconstrução densa, e determine se o par de imagens tem condições de ser retificado satisfatoriamente pela forma planar. Caso a resposta seja positiva, este método será responsável pela retificação. Se não, o método cilíndrico será acionado.

6.1. TRABALHOS FUTUROS

Mesmo a técnica híbrida não seria rápida o suficiente caso o desejado fosse realizar uma reconstrução densa em tempo real. Para que este requisito se torne possível é necessário o uso de computação paralela. Atualmente, uma forma comum de conseguir este tipo de computação é utilizar GPUs, sigla em inglês para unidades gráficas de processamento, que suportam a arquitetura de computação paralela CUDA (*Compute Unified Device Architecture*) [27].

O desenvolvimento deste algoritmo de forma paralela é possível devido ao fato da retificação cilíndrica ser um processo onde cada linha epipolar da imagem é manipulada de forma independente das demais. Desta forma, pode-se fazer uso desta arquitetura para que todas as linhas sejam manipuladas simultaneamente. De fato, é possível que se consiga reduzir o tempo de execução para a casa dos milissegundos, tornando a retificação viável em tempo real.

Como a retificação de imagens é o primeiro passo da reconstrução densa, é natural que uma das próximas atividades seja a implementação das etapas seguintes do *dense matching*.

Por outro lado, o processo de reconstrução densa também pode ser realizado sem retificar as imagens. Durante a fase de pesquisa deste trabalho foram encontradas referências, tais como [28] e [29], que mostram ser possível encontrar o mapa de profundidade, e posteriormente a posição tridimensional de todos os pontos de duas imagens, utilizando um descritor denso. Como o próprio nome já diz, esse descritor, chamado DAISY, descreve unicamente cada *pixel* da imagem. Desta forma, é possível encontrar pontos correspondentes entre duas imagens apenas analisando esses dados. Assim, dois pontos serão classificados como correspondentes se os seus descritores forem iguais.

Durante a avaliação do DAISY, foi constatado que esta técnica, apesar de recente, consegue resultados para a reconstrução densa tão precisos quanto os obtidos usando os métodos já tradicionais, como os descritos neste trabalho. Além disso, os autores afirmam que, usando o DAISY, um mapa de profundidade para uma imagem quadrada de tamanho 1024 pode ser encontrado em cinco segundos, o que é bem mais rápido do que somente a retificação de imagem. Entretanto, foi preferido estudar e implementar as formas tradicionais por causa do conteúdo matemático extremamente denso desta abordagem, que levaria bastante tempo para ser totalmente compreendido. Além disso, existe também a perspectiva da reconstrução densa convencional ser implementada fazendo uso de computação paralela, reduzindo muito o tempo de execução, como já mencionado.

7. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Microsoft. Photosynth: Your photos, automatically in 3D. [Online]. HYPERLINK "http://photosynth.net/" http://photosynth.net/

[2] Yi Ma, Stefano Soatto, Jana Kosecká e S. Shankar Sastry, *An Invitation to 3D Vision*.: Spring, 2006.

[3] Marc Pollefeys, "Self-Calibration and Metric 3D Reconstruction From Uncalibrated Image Sequences," Katholieke Universiteit Leuven, Heverlee, 1999.

[4] Luigi Cremona, *Elements of Projective Geometry*, 3rd ed. Estados Unidos: Dover Publications.

[5] Marc Pollefeys, "Visual 3D Modeling from Images," University of North Carolina, North Carolina, Tutorial Notes 2000.

[6] Agma Tbaina, Afonso Prado e Josiane Bueno, "3D Reconstruction of Tomographic Images Applied to Largely Spaced Slices," *Journal of Medical Systems*, 1997.

[7] Ildiko Suveg e George Vosselman, "3D Building Reconstruction by Map Based Generation and Evaluation of Hypotheses," in *British Machine Vision Conference*, 2001.

[8] Severino Neto, Márcio Bueno, Thiago Farias, João Pualo Lima, Veronica Teichrieb, Judith Kelner e Ismael Santos, "Experiences on the Implementation of a 3D Reconstruction Pipeline," *International journal of modeling and simulation for the petroleum industry*, 2008.

[9] GRVM. GRVM - Virtual Reality and Multimedia Research Group. [Online]. HYPERLINK "http://www.gprt.ufpe.br/~grvm/" http://www.gprt.ufpe.br/~grvm/

[10] CIn..: CIn - Centro de Informática UFPE . [Online]. HYPERLINK "http://www.cin.ufpe.br" http://www.cin.ufpe.br

[11] UFPE. Universidade Federal de Pernambuco. [Online]. HYPERLINK "http://www.ufpe.br" http://www.ufpe.br

[12] Anthony Hii, Christopher Hann, Geoffrey Chase e Eli Van Houten, "Fast normalized cross correlation for motion tracking using basis functions," *Computer methods and programs in biomedicine* , 2006.

[13] Carlo Tomasi e Takeo Kanade, "Detection and Tracking of Point Features," Carnegie Mellon University, 1991.

[14] Richard I. Hartley, "In Defense of the Eight-Point Algorithm," in *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 1997.

[15] Marc Pollefeys. Computer Vision (comp256). [Online]. HYPERLINK "http://www.cs.unc.edu/Research/vision/comp256/" http://www.cs.unc.edu/Research/vision/comp256/

[16] Richard Hartley e Andrew Zisserman, *Multiple View Geometry in computer vision*. Cambridge: Cambridge University Press, 2003.

[17] Jean-Daniel Boissonnat, "Representing 2d and 3d shapes with the Delaunay triangulation," in *International Conference on Pattern Recognition*, 1894.

[18] Raphaëlle Chaine, "A geometric convection approach of 3-D reconstruction," in *ACM International Conference Proceeding Series*, 2003.

[19] Daniel Oram, "Rectification for Any Epipolar Geometry," in *British Machine Vision Conference*, Manchester, 2001.

[20] Andrea Fusiello, Emanuele Trucco e Alessandro Verri, "A compact algorithm for rectification os stereo pairs," in *Machine Vision and*

Applications, 2000.

[21] Sébastien Roy, Jean Meunier e Ingemar Cox, "Cylindrical rectification to minimize epipolar distortion," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, 1997.

[22] Marc Pollefeys, "A simple and eficiente rectification method for general motion," in *International Conference on Computer Vision*, 1999.

[23] Edward Angel, *Interactive Computer Graphics A Top-Down Approach with OpenGL*.: Addison-Wesley, 2006.

[24] MathWorks. MATLAB - The Language Of Technical Computing. [Online]. HYPERLINK "http://www.mathworks.com/products/matlab/" http://www.mathworks.com/products/matlab/

[25] Manolis Lourakis e Antonis Argyros. Efficient, Causal Camera Tracking In Unprepared Environments. [Online]. HYPERLINK "http://www.ics.forth.gr/~lourakis/camtrack/" http://www.ics.forth.gr/~lourakis/camtrack/

[26] vision.middlebury.edu. Multi-view stereo evaluation page. [Online]. HYPERLINK "http://vision.middlebury.edu/mview/" http://vision.middlebury.edu/mview/

[27] NVidia. CUDA Zone -- The resource for CUDA developers. [Online]. HYPERLINK "http://www.nvidia.com/object/cuda_home.html" http://www.nvidia.com/object/cuda_home.html

[28] Engin Tola, Vincent Lepetit e Pascal Fua, "A fast local descriptor for dense matching," *Proceedings of Computer Vision and Pattern Recognition*, Junho 2008.

[29] Engin Tola, Vincent Lepetit e Pascal Fua, "DAISY: an efficient dense descriptor applied to wide baseline stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Março 2009.