Relatório sobre os Resultados obtidos através do uso dos algoritmos SIFT e RANSAC para Reconstrução de um Objeto a partir de uma Nuvem de Pontos

Gustavo Moreira
PUC-Rio, Departamento de Informática
Rua Marquês de São Vicente, 225 – Gávea – 22453-900, Rio de Janeiro, RJ, Brasil
gmoreira@inf.puc-rio.br

Resumo

Neste relatório apresentamos um resumo sobre os resultados obtidos em um trabalho que tinha como objetivo a reconstrução de um objeto a partir de sua nuvem de pontos. O algoritmo SIFT foi utilizado para encontrar a correspondência de pontos-chave entre as imagens do objeto, enquanto o RANSAC foi utilizado para encontrar a matriz de homografia entre os pares correspondentes entre as imagens e reduzir a quantidade de falsas correspondências entre os pontos-chave das imagens. Uma descrição resumida de ambos os algoritmos, SIFT e RANSAC, é também fornecida neste artigo.

1. Introdução

A correspondência de imagens é fundamental em diversos problemas de visão computacional como reconhecimento de objetos, reconhecimento de cenas, montagem automática de mosaicos, obtenção da estrutura 3D de múltiplas imagens, correspondência estéreo e perseguição de movimentos. Uma abordagem para se trabalhar com correspondência de imagens é se usar descritores locais para se representar uma imagem. Descritores são vetores de características de uma imagem ou de determinadas regiões de uma imagem e podem ser usados para se comparar regiões em imagens diferentes. Este vetor de características é normalmente formado por descritores locais ou globais. Descritores locais computados em pontos de interesse provaram ser bem sucedidos em aplicações como correspondência e reconhecimento de imagens. Descritores são distintos, robustos à oclusão e não requerem segmentação.

Existem diversas técnicas para se descrever regiões locais em uma imagem. O mais simples descritor é um vetor com as intensidades dos pixels da imagem. A medida de correlação cruzada pode ser então usada para computar a similaridade entre duas regiões. Porém, a alta dimensionalidade de tal descritor aumenta a complexidade computacional da comparação. Então, esta técnica é principalmente usada para se encontrar correspondências ponto a ponto entre duas imagens. A vizinhança de um ponto também pode ser escalada de modo a reduzir sua dimensão. Outro descritor simples é a distribuição de intensidades de uma região representada por seu histograma.

Trabalhos recentes na literatura têm se concentrado em fazer descritores invariáveis a transformações nas imagens.

2. SIFT

SIFT (“Scale Invariant Feature Transform”) é uma técnica, definida por David Lowe, para o processamento de imagens que permite a detecção e extração de descritores locais, razoavelmente invariáveis a mudanças de iluminação, ruído de imagem, rotação, escala e pequenas mudanças de perspectiva. Estes descritores podem ser utilizados para se fazer a correspondência de diferentes visões de um objeto ou cena.

Descritores obtidos com a técnica SIFT são altamente distintos, ou seja, um determinado ponto pode ser corretamente encontrado com alta probabilidade em um banco de dados extenso com descritores para diversas imagens.

Um aspecto importante da técnica SIFT é a geração de um número grande de descritores que conseguem cobrir densamente uma imagem quanto a escalas e localização. A quantidade de descritores é particularmente importante para o reconhecimento de objeto, onde a capacidade de se encontrar pequenos objetos em ambientes desordenados requer ao menos 3 pontos encontrados em comum para uma identificação confiável.

A obtenção de descritores SIFT é feita através das seguintes etapas:

- **Detecção de extremos**: Nesta primeira etapa é feita procura para todas as escalas e localizações de uma imagem. Isto é feito utilizando-se a diferença de filtros gaussianos de modo a se identificar pontos de interesse invariáveis à escala e rotação;

- **Localização de pontos chave**: Para cada localização em que foi detectado um extremo, um modelo detalhado é ajustado de modo a se determinar localização e escala. Pontos chaves, ou pontos de interesse, são então selecionados baseando-se em medidas de estabilidade;

- **Definição de orientação**: É definida a orientação de cada ponto chave através dos gradientes locais da imagem. Toda operação a partir de então será feita com relação a dados da imagem transformados em relação à orientação, escala e localização de cada ponto chave. Desta maneira se obtém invariância a estas transformações;

- **Descritor dos pontos chaves**: Nesta etapa é feita a construção dos descritores ao se medir Gradientes locais em uma região vizinha a cada ponto de interesse. Estas medidas são então transformadas para uma representação que permite níveis significativos de distorção e mudança na iluminação;

Em tarefas de comparação de imagens e reconhecimento, descritores SIFT são extraídos das imagens para então poderem ser comparados. Na próxima seção será descrito o primeiro passo da obtenção de descritores SIFT.

Detecção de Extremos

A primeira etapa da técnica SIFT é detectar extremos (máximos e mínimos) em uma pirâmide da imagem convoluída com a função Diferença de Gaussiana. Pontos chave correspondem a estes extremos para diferentes escalas.

A convolução de uma função f(x,y) com uma função h(x,y) é dada por:

$$f(x,y) * h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n)$$

Onde x varia de 1 a M e y varia de 1 a N.

Um filtro Gaussiano passa baixa é dado pela convolução de uma imagem I com a função G:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(x^2+y^2\right)/2\sigma^2}$$

A função DoG ("Difference of Gaussian") é dada pela diferença de imagens filtradas em escalas próximas separadas por uma constante k. A função DoG é definida por:

$$\mathrm{DoG} = G(x,y,k\sigma) - G(x,y,\sigma)$$

A convolução de uma imagem com o filtro DoG é dada por:

$$D(x,y,\sigma) = \left(G(x,y,k\sigma) - G(x,y,\sigma)\right) * I(x,y)$$
$$= L(x,y,k\sigma) - L(x,y,\sigma)$$

Ou seja, é a diferença entre imagens borradas por um filtro gaussiano em escalas σ e k σ. Este filtro consegue detectar variações de intensidade na imagem, tais como contornos. Percebe-se que variando σ, é possível encontrar descritores para variações em diferentes escalas espaciais.

Deseja-se construir s intervalos, onde cada intervalo representa uma imagem filtrada por DoG intervalar entre duas outras. Para se construir s intervalos serão necessárias s+3 imagens na pilha apresentada pelas imagens superiores da Figura 1. A imagem inicial é convoluída progressivamente com funções gaussianas para produzir mais s+2 imagens separadas por um fator constante k. A imagem é inicialmente filtrada por filtro Gaussiano com escala σ. A partir de então, são geradas imagens que são progressivamente convoluídas. Cada nova imagem é filtrada com escala k vezes a escala utilizada anteriormente. Para cada duas imagens, pode-se produzir a diferença de Gaussianas D através da subtração de duas imagens consecutivas na pilha de imagens L.

Para exemplificar, imagine que se deseja gerar apenas um intervalo. Serão necessárias, então, quatro imagens na pilha superior (a pilha das imagens filtradas com Gaussianas). Estas imagens serão:

- A imagem original;
- A imagem original filtrada por Gaussiana com escala σ;
- Outras duas imagens filtradas com escalas multiplicadas por k: kσ e k2σ;

David Lowe considera que é necessário fazer a convolução da imagem até 2 σ para ser possível a construção de descritores invariáveis à escala. Portanto, para se gerar s intervalos é definido:

$$k = 2^{1/s}$$

Desta maneira, teremos s intervalos produzidas por DoG, sendo que o primeiro é dado por D(x,y, σ) e a ultima imagem da pilha de DoG dada por D(x,y,2 σ). Para melhor entendimento, perceba na figura 1, a primeira imagem acima à esquerda é I(x,y) e a última imagem acima é L(x,y,k2 σ). A primeira imagem abaixo é D(x,y,0) e a última é D(x,y,2 σ). O intervalo é dado por D(x,y, σ). O processo apresentado gera o que é chamado de uma oitava. Este processo é repetido para um número desejado de oitavas. Cada oitava representa um conjunto de imagens L e D para a imagem reescalada com diferentes amostragens.

Isto funciona da seguinte forma: quando uma oitava tiver sido processada, a imagem Gaussiana que possui 2 σ (corresponde à penúltima imagem da pilha superior na Figura 1) é re-amostrada para a metade de seu tamanho. Esta será a primeira imagem da próxima oitava. Cada oitava produz o mesmo número de intervalos.

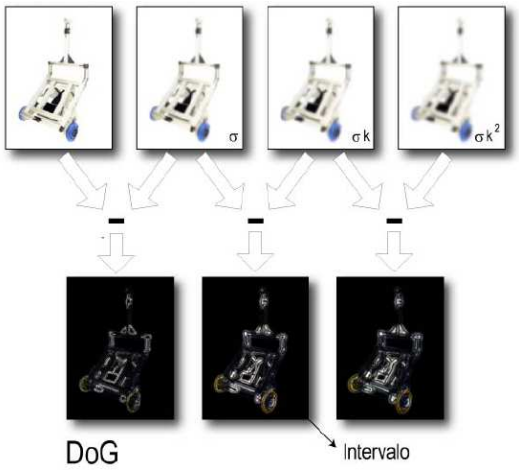

Figura 1.

A partir de agora será feita a detecção de extremos em cada intervalo de cada oitava. Os extremos são dados por valores locais de máximo ou mínimo para cada D(x ,y, σ) que corresponda a um intervalo. Cada ponto é comparado aos seus oito vizinhos na imagem atual, mais seus nove vizinhos na escala superior e nove vizinhos na escala inferior.

As escalas superior e inferior são correspondentes às imagens vizinhas em uma mesma oitava para a pilha de imagens DoG. Não confunda escala superior e inferior com oitavas onde a amostragem das imagens gera imagens em escalas diferentes. Quando se diz escala superior e inferior aqui, está se fazendo referência à σ.

O procedimento de detecção está exemplificado na Figura 3. No exemplo, o ponto marcado como X é comparado com seus vizinhos marcados como O. As 3 imagens DoG apresentadas são 3 intervalos vizinhos em uma pilha.

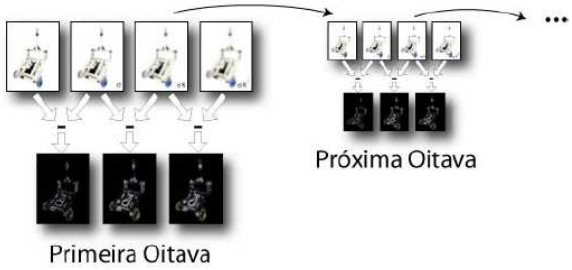

Figura 2

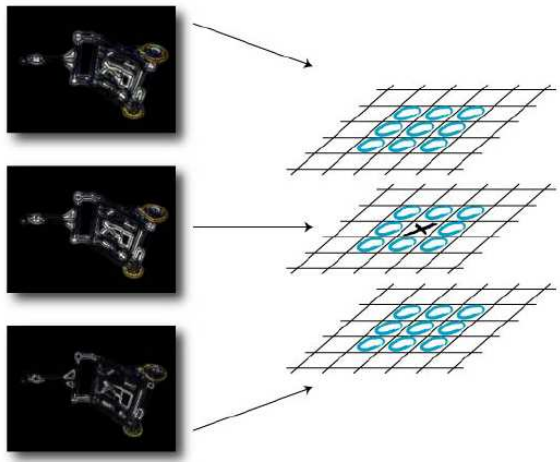

Figura 3

Localização Exata de Pontos Chave

Todos os pontos detectados como extremos são possíveis pontos chave.

Deseja-se agora calcular a localização e escala Gaussiana detalhadas de cada um destes pontos. Recalcular a localização e escala interpolada dos pontos de máximo traz melhoria para a técnica. A localização dos pontos chaves como será apresentada é especialmente importante para as últimas oitavas, porque o espaçamento amostral destas representa grandes distâncias na imagem base.

O método consiste em enquadrar uma função quadrática 3D do ponto de amostragem local de modo a determinar uma localização interpolada do máximo.

Para cada ponto analisado é utilizada uma expansão de Taylor da função D(x,y, σ) transladada de modo que a origem desta expansão esteja localizada no ponto:

$$D(\bar{x}) = D + \frac{\partial D^T}{\partial \bar{x}}\bar{x} + \frac{1}{2}\bar{x}^T \frac{\partial^2 D}{\partial x^2}\bar{x} \ldots$$

Onde:

$$D = D(x, y, \sigma)$$

$$D(\bar{x}) = D(x + x', y + y', \sigma + \sigma')$$

Esta equação deve ser entendida da seguinte maneira, D e suas derivadas são avaliadas a partir do ponto analisado e $\bar{x} = (x', y', \sigma')^T$ é o offset em relação a este ponto. Ou seja, D é o valor da função D(x,y,σ) no ponto avaliado, x é o offset em relação a este ponto e $D(\bar{x})$ é a aproximação do valor de D(x,y,σ) interpolado para um ponto transladado com offset x.

Os coeficientes quadráticos são computados aproximando-se as derivadas através das diferenças entre pixels das imagens já filtradas.

A localização "sub-pixel / sub-escala" do ponto de interesse é dada pelo extremo da função apresentada na equação anterior. Esta localização, $\hat{x}$, é determinada ao se fazer a derivada segunda da equação anterior com relação à x e igualando o resultado à zero. Isto é feito como a seguir:

$$\frac{\partial D(\hat{x})}{\partial \bar{x}} = \frac{\partial D^T}{\partial \bar{x}} + \frac{\partial D}{\partial \bar{x}^2}\hat{x} = 0$$

Perceba que esta derivada usa a expansão de Taylor até $\frac{\partial D}{\partial \bar{x}}$. Tem-se então a posição do extremo dada por:

$$\hat{x} = -\frac{\partial^2 D^{T-1}}{\partial \bar{x}^2}\frac{\partial D}{\partial \bar{x}}$$

O resultado é um sistema linear 3x3 que pode ser resolvido com custo mínimo. Caso $\hat{x}$ seja maior que 0.5 em alguma dimensão. Isto significa que o extremo se aproxima mais de outro ponto. Neste caso, o ponto é re-alocado e a interpolação é realizada para este novo ponto. O offset $\hat{x}$ final é adicionado à localização do ponto analisado para se chegar à interpolação estimada da localização do extremo.

A localização estimada deverá ser usada a partir de então nos procedimentos que seguirão.

O valor da função no extremo, D ($\hat{x}$), é utilizado para se rejeitar extremos instáveis com baixo contraste. Substituindo-se obtemos:

$$D(\hat{x}) = D + \frac{1}{2}\frac{\delta D^T}{\delta \bar{x}}\hat{x}$$

É aconselhável por Lowe que se rejeitem valores de D($\hat{x}$) inferiores a um determinado valor. É aconselhado trabalhar-se com o valor 0.03(assumindo-se que os pixels da imagem estejam entre [0,1]).

Aqui não é refinada a posição como apresentado, porém são descartados valores de |D(x,y, σ)| inferiores a determinado limiar.

Alem do procedimento apresentado para se descartar pontos, Lowe ainda aponta que a função DoG possui resposta forte ao longo de arestas, mesmo que a localização ao longo da borda seja mal determinada. Isto faz com que estes pontos sejam instáveis para ruído em até pequenas quantias.

Atribuição da Orientação dos Descritores

Ao se atribuir uma orientação para cada ponto chave, podem-se representar os descritores em relação a esta orientação, conseguindo-se assim invariância quanto à rotação. O método utilizado para se atribuir esta orientação é apresentado como se segue.

A escala Gaussiana σ é utilizada para se escolher a imagem filtrada L, com a escala mais próxima, e de oitava referente ao ponto avaliado. Dessa maneira, todas os cálculos passam a ser feitos com invariância à escala.

Para cada ponto de cada imagem L(x,y, σ) intervalar, referente às escalas e oitavas utilizadas, são calculados os gradientes. Magnitude m(x,y) e orientação θ(x,y) são calculados como se segue:

$$m(x,y) = \sqrt{\begin{matrix}(L(x+1,y) - L(x-1,y))^2 \\ +(L(x,y+1) - L(x,y-1))^2\end{matrix}}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{(L(x,y+1) - L(x,y-1))}{(L(x+1,y) - L(x-11,y))}\right)$$

Observe que σ não aparece nas equações. Isto foi feito para simplificar, pois o processamento é feito para cada imagem L. Somente as imagens correspondentes a intervalos precisam ser processadas.

Agora, monta-se um histograma das orientações para pixels em uma região ao redor do ponto chave. O histograma é uma função discreta h_θ um determinado

número de valores discretos de θ (Lowe sugere 36) cobrindo os 360º de orientações.

Cada ponto na vizinhança do ponto chave é adicionado ao histograma para até dois θ's discretos mais próximos de sua orientação com uma serie de pesos.

O primeiro peso é dado pela distância entre a orientação e θ discreto normalizado pelas distâncias entre θ's discretos. Este peso é dado por:

$$\alpha = \begin{cases} d/i, d < i \\ 0, d > i \end{cases}$$

Onde d é a distância absoluta em graus entre a orientação do ponto e θ discreto, e i é o intervalo em graus entre θ's discretos.

Por exemplo, para h_θ com θ dado por 0º,10º, 20º... 350º, ou seja, com intervalos de 10º, um ponto com orientação de 15º seria acrescentado em h10 e h20. Como a distância da orientação para 10º e 20º é de 5º, o peso utilizado para adicionar este ponto em para h10 e h20 é dado por 5/10, ou seja, a distância sobre o intervalo entre θ's para h_θ.

O segundo peso é dado pela magnitude m(x,y) de cada ponto adicionado a O último peso é dado por uma janela gaussiana circular com σ' com valor 1.5 vezes maior que a escala σ do ponto chave. Esta janela é definida pela função gaussiana:

$$g(\Delta x, \Delta y, \sigma') = \frac{1}{2\pi\sigma'^2} e^{-\left(\Delta x^2 + \Delta y^2\right)/2\sigma'^2}$$

$$\sigma' = \sigma$$

Onde Δx e Δy são as distâncias entre cada ponto verificado e o ponto chave.

Por fim, h_θ é atualizado com estes pesos, para cada ponto na vizinhança localizado em (x,y), da seguinte forma:

$$h_\theta' = h_\theta + \alpha \cdot m(x,y) \cdot g(\Delta x, \Delta y, \sigma')$$

Percebe-se que não é necessário fazer a atualização de h_θ para todos os pontos da imagem porque a função $g(\Delta x, \Delta y, \sigma')$ retorna valores muito baixos (aproximadamente zero) para a grande maioria dos pontos.

Picos na orientação do histograma correspondem a direções dominantes para os gradientes locais. O maior pico no histograma e aqueles acima de 80% do valor do maior pico são usados para se definir a orientação de cada ponto chave.

Portanto, para localizações com múltiplos picos de magnitude similar, são criados diferentes pontos chaves na mesma localização, mas com diferentes orientações.

Para se definir com maior precisão à orientação, uma parábola é interpolada entre os 3 valores do histograma próximos de cada pico, e então é interpolada a posição do pico.

Construção do Descritor Local

Até então, para cada oitava, foram escolhidos pontos chaves para localizações, escala σ e orientação definidos. A etapa atual consiste em computar o descritor que represente as regiões relativas aos pontos chaves. Os procedimentos a seguir são feitos normalizados em relação à orientação definida na seção anterior para cada ponto chave.

Para cada ponto chave, a construção do descritor é feita através dos seguintes passos:

- Escolhe-se a imagem filtrada L referente à escala σ e oitava relativas ao ponto chave;
- De modo a se conseguir invariância, as coordenadas dos pontos vizinhos ao descritor e das orientações dos gradientes destes pontos são giradas em relação ao ponto chave de acordo com a orientação definida na seção anterior;
- Uma função gaussiana é utilizada como peso para se ajustar as magnitudes de cada ponto na vizinhança do ponto chave. σ' é escolhido igual à metade da largura da janela em que será calculado o descritor;
- São definidas n x n regiões, com k x k pixels cada, ao redor da localização do ponto chave. Geralmente n = k = 4.
- Para cada região, é feito um histograma h_θ. para 8 direções, como na 4. Este histograma é feito com as magnitudes dos pixels pertencentes a cada região. O peso referente à magnitude de cada pixel foi atenuado pela função gaussiana como já ajustado. Percebe-se que a função gaussiana não é aplicada de modo idêntico ao na seção anterior. É utilizado um peso α para interpolar a direção relativa no histograma.
- O descritor é então representado pelos histogramas das regiões. A Figura 5 exemplifica como fica o descritor para 2 x 2 regiões (n = 2 e k = 4);
- O descritor é representado por um vetor, onde cada valor do vetor é referente a uma das direções de um dos histogramas. Para n e k iguais a 4, o vetor tem tamanho 128.
- Para que o descritor tenha invariância à iluminação, este é normalizado. Após a normalização, todos os valores acima de um determinado limiar são ajustados para este limiar. Isto é feito para que direções com magnitude muito grande não dominem a representação do descritor. Lowe sugere usar limiar 0.2. Por fim, o vetor é normalizado novamente.

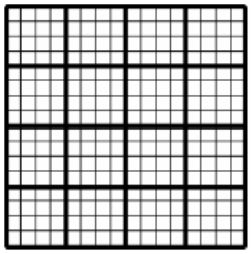

Figura 4. Regiões com n=4 e k=4

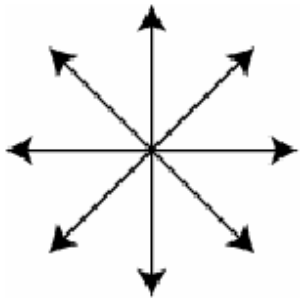

Figura 5. Regiões do Histograma

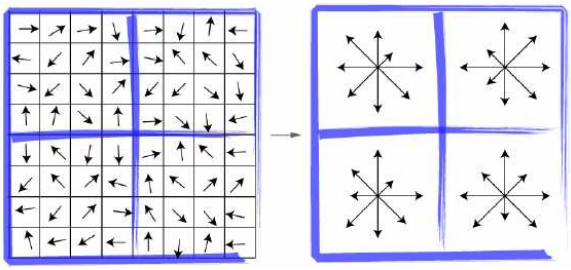

Figura 6. Construção do Descritor

O descritor está construído. Para cada imagem, são construídos diversos descritores, cada um referente a um ponto chave. Quando se aplica a técnica SIFT em uma imagem, tem-se como resultado, portanto, um conjunto de descritores. Estes descritores podem ser então, usados para se fazer a correspondência da imagem em outra imagem.

Encontrando os Pontos em Comum

Para se encontrar a correspondência entre duas imagens, devem-se encontrar pontos em comum entre as duas. Quando se trabalha com a técnica SIFT, pontos de interesse são detectados pelo método e representados em descritores. Tendo-se descritores de duas imagens, a tarefa de se encontrar a correspondência de uma imagem em outra é resumida por se encontrar entre os descritores de uma imagem, os melhores candidatos a serem seus equivalentes na outra imagem.

Portanto, dadas duas imagens I1 e I2, a tarefa de se encontrar a correspondência de I1 em I2 pode ser definida como se segue.

Os descritores são respectivamente definidos por di1 e dj2, onde i e j são aos índices para cada um dos descritores de cada imagem e k é o tamanho de cada descritor:

$$di_1 = (m_1 i_1, m_1 i_2, m_1 i_3 \cdots, m_1 i_k)$$

$$dj_2 = (m_2 j_1, m_2 j_2, m_2 j_3 \cdots, m_2 j_k)$$

A magnitude de cada valor dos vetores di1 e dj2 é dada por $m_a i_b$, onde a representa a qual imagem se refere o descritor, i é o índice do descritor e b é o índice de cada magnitude dentro do vetor.

A correspondência é feita achando-se os descritores dj2 que mais se assemelham aos descritores di1, encontrando-se as falsas equivalências e eliminando-as e por fim, encontrando-se a transformação de I1 para I2.

A tarefa de se encontrar o melhor candidato dj2 para determinado di1 é feita procurando-se o vizinho mais próximo ou “nearest neighbor” de di1 entre todos os possíveis candidatos, ou seja, para todo o índice j. Quando se procura classificar uma imagem em um extenso banco de dados de descritores para vários objetos, a busca exaustiva de vizinho mais próximo pode ser demorada e para tal existem diversas técnicas para se acelerar a busca. Porém, para o caso de se comparar duas imagens, a busca exaustiva não exige processamento pesado e, portanto, foi a escolhida.

O vizinho mais próximo de di1 para i dado é definido por dj2 que possua a menor distância euclideana em relação à di1. Ou seja, deseja se encontrar j que minimize a função:

$$|di_1 - dj_2| = \sqrt{\begin{matrix}(m_1 i_1 - m_2 j_1)^2 + (m_1 i_2 - m_2 j_2)^2 + \ldots \\ +(m_1 i_k - m_2 j_k)^2\end{matrix}}$$

Isto é feito para todo i de modo a serem encontrados todos os pares de descritores correspondentes. Perceba que muitos dos pares encontrados correspondem a falsas equivalências, portanto, as correspondências serão refinadas e falsos pares descartados.

3. RANSAC

RANSAC (“Random Sample Consensus”) é um paradigma para ajuste robusto de um modelo de dados experimentais. Ele é geralmente usado em visão

computacional, como por exemplo, para resolver simultaneamente o problema de correspondência entre pontos de duas imagens e estimar a matriz fundamental relacionada ao par de imagens estéreo.

Uma vantagem do RANSAC é a sua habilidade de realizar a estimativa de parâmetros de um modelo de forma robusta, ou seja, ele pode estimar parâmetros com um alto grau de acerto mesmo quando um número significativo de "outliers" esteja presente nos dados analisados. Uma desvantagem do algoritmo é que não há um limite superior de tempo para que ele possa computar tais parâmetros. Quando um limite superior é usado (número máximo de iterações) a solução obtida pode não ser a melhor existente.

Os métodos clássicos procuram utilizar o maior número de pontos para obter uma solução inicial e, então, eliminar os pontos inválidos. O RANSAC, ao contrário desses métodos, utiliza apenas o número mínimo e suficiente de pontos necessários para uma primeira estimativa, aumentando o conjunto com novos pontos consistentes sempre que possível.

Dado um modelo com parâmetros $\vec{x}$, deseja-se estimá-los. Para tal, é assumido:

- Os parâmetros podem ser estimados a partir de um número N de itens em um conjunto de dados conhecido;
- Existe um total de M itens no conjunto de dados;
- A probabilidade de um dado selecionado aleatoriamente fazer parte de um bom modelo é dada por p_g;
- A probabilidade de que o algoritmo termine sem que se encontre um bom modelo é dada por p_{falha};

O algoritmo é então executado através das seguintes etapas:

1. N itens são escolhidos de modo aleatório;
2. A partir dos itens escolhidos, $\vec{x}$ é estimado;
3. Encontra-se o número de itens que se enquadram ao modelo para determinada tolerância especificada. Este número é chamado de K;
4. Caso K seja grande o suficiente, para um limiar escolhido, o algoritmo termina e foi bem sucedido;
5. O algoritmo é repetido de 1 a 4 um número L de vezes;
6. Caso o algoritmo não tenha terminado após L tentativas, o algoritmo falhou;

L pode ser encontrado através das seguintes maneiras:

- p_{falha} = Probabilidade de L falhas consecutivas;
- p_{falha} = (Probabilidade de que determinado dado seja falho)*L;
- p_{falha} = (1 – Probabilidade de sucesso)*L;
- p_{falha} = (1 – (Probabilidade de que um item caiba no modelo)*N)*L;
- p_{falha} = (1 – (pg) *N) *L;

Para o problema específico de remoção de "outliers" na correspondência de imagens estéreo, a Matriz Fundamental (F) pode ser utilizada da seguinte forma:

◦ Selecionar randomicamente um subconjunto de oito pontos de correlacionados, retirados do conjunto total de pontos correlacionados (através de SIFT, por exemplo).

◦ Para cada subconjunto, indexado por j, calcular a matriz fundamental Fj através do algoritmo dos oito pontos.

◦ Para cada matriz Fj computada, determinar o número de pontos com distância até a linha epipolar, ou residual, menor que um limiar.

◦ Selecionar a matriz F que apresenta o maior número de pontos com residual inferior ao máximo definido.

◦ Recalcular a matriz F considerando todos os pontos "inliers".

$$\tilde{m}^T F \tilde{m}' = 0$$

4. Resultados Obtidos e Conclusão

O trabalho desenvolvido por Hildebrando Tranning teve como objetivo implementar uma aplicação que gere uma nuvem de pontos para posterior geração manual de uma malha de triângulos de um objeto 3D, tendo como entra um vídeo ou um conjunto de imagens, e como saída, a nuvem de pontos do objeto.

Para tanto, as principais etapas do programa são:

- Extração de frames – Quando utilizado um vídeo, Trannin sugere um vídeo não muito distante do objeto em questão, próximo de 5 graus, pois em distâncias maiores haveria perda da detecção de características do objeto;

- Detecção de características – Utilização do algoritmo SIFT para a extração dos descritores de características do objeto;
- Calibração da câmera – Utilizado o algoritmo proposto por Tsai e o padrão de Xadrez;
- Correspondência entre essas características – Utiliza facilidades oferecidas pela detecção de características ter sido feito com SIFT. Também utiliza o RANSAC para descarte dos “outliers”, ou seja, dos falsos “matches”;
- Criação de retalhos entre par de frames;
- Alinhamento dos retalhos para geração da nuvem de pontos através de triangulação.

Abaixo são exemplificados a detecção dos descritores locais em dois frames subseqüentes.

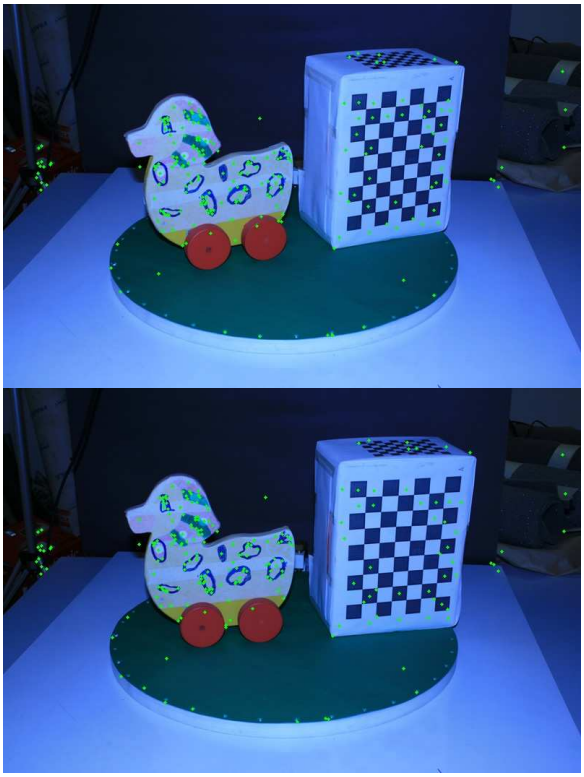

Figura 7

Entre as duas imagens da figura 7, é possível notar que alguns pontos encontrados na primeira imagem não correspondem a pontos encontrados na segunda imagem. Com o intuito de minimizar tais pontos, Trannin utilizou o RANSAC. O resultado obtido por ele é exibido na figura 8.

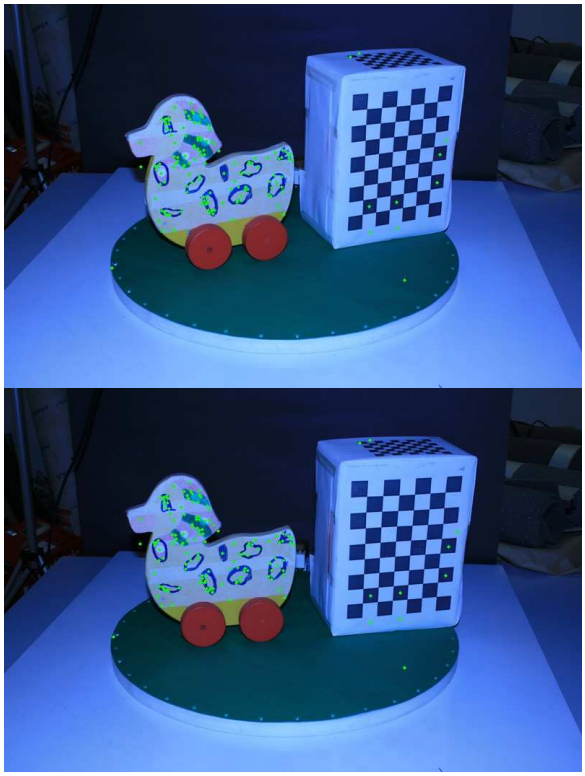

Figura 8

Observando a figura 8, nota-se que vários pontos foram excluídos da cena após o filtro de “outliers” executado pelo RANSAC, principalmente aqueles pontos encontrados dentro do padrão do tabuleiro de xadrez. Este fato ocorreu devido ao valor de um dos parâmetros utilizados por Trannin ao executar o algoritmo RANSAC. O parâmetro “err_tol” existente na função “ransac_xform” faz com que pontos de correspondências encontrados dentro deste limite de distância um dos outros sejam considerados como “inliers” para uma dada matriz de transformação. O que aconteceu para ocorrer esta elevada redução de matches dentro do padrão de tabuleiro de xadrez foi um valor muito baixo para esta distância, com o valor utilizado sendo igual a 3. Ao se alterar o valor deste parâmetro para 8 por exemplo, é possível aumentar a flexibilização do algoritmo RANSAC com o intuito de considerar mais correspondências da cena como sendo “inliers” de maneira correta, conforme mostrado pelas imagens da figura 9.

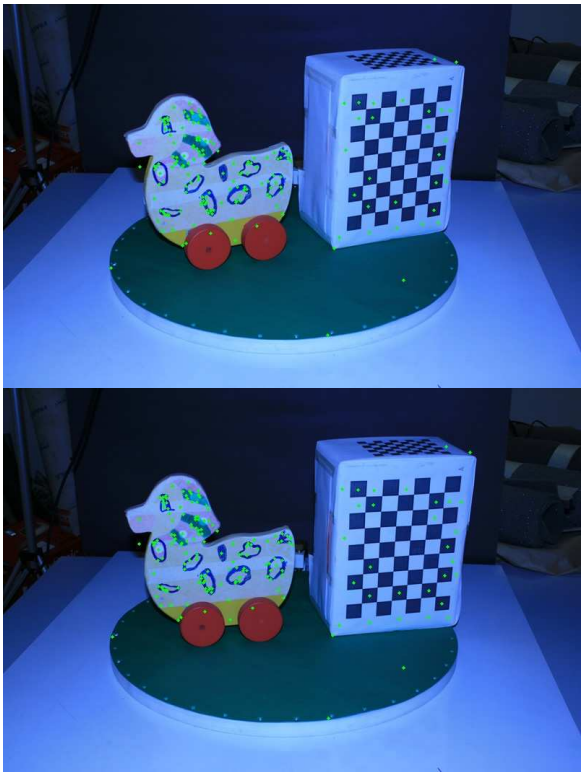

Figura 9

Portanto, é possível concluir que o algoritmo RANSAC de fato é capaz de reduzir a ocorrência de falsas correspondências entre descritores locais encontrados pelo algoritmo SIFT. Entretanto, é necessário permitir que o algoritmo RANSAC tenha uma configuração de parâmetros tal que permita certa flexibilização com o intuito de não reduzir de maneira muito elevada a quantidade de correspondências entre duas imagens. Desta forma, com o RANSAC encontrando um número maior de pontos, a nuvem de pontos gerada ao final da aplicação permitirá a reconstrução da malha do objeto de forma mais completa e eficiente.

5. Referências

[1] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. Int. Journal of Computer Vision, 60(2):91–110, 2004.

[2] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM, 24(6):381–395, 1981.

[3] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. IEEE Transactions on pattern Analysis and Machine Intelligence, 6:13–27, 1984.

[4] H. Trannin. Reconstrução 3D. Trabalho da Disciplina "Visão Computacional e Realidade Aumentada". Prof. Marcelo Gattass. http://www.tecgraf.puc-rio.br/~mgattass/ra/trb08/HildebrandoTrannin/ Acessado em Dezembro/2008.