



**UNIVERSIDADE FEDERAL  
DE SANTA CATARINA**

SERVIÇO PÚBLICO FEDERAL  
**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS BLUMENAU**

*Rua João Pessoa, 2750, Velha, Blumenau – SC.*  
[www.blumenau.ufsc.br/blumenau@contato.ufsc.br](http://www.blumenau.ufsc.br/blumenau@contato.ufsc.br)

## **VISÃO COMPUTACIONAL**

### **Relatório A1**

Gabriel Alves Silvestre

Ronaldo William Baggio de Oliveira

Blumenau

Abril, 2018

## **1 INTRODUÇÃO**

O presente trabalho trata da descrição técnica do primeiro trabalho apresentado para a disciplina de Visão Computacional, neste documento será apresentada parte do embasamento teórico do trabalho que tem como tema Homografia.

O objetivo deste trabalho é elaborar um algoritmo capaz de construir uma única imagem panorâmica a partir de um conjunto de cinco imagens de uma edificação. Tal algoritmo deve corrigir a perspectiva das cinco imagens, visto que as imagens foram obtidas de maneiras diferentes, além disso deve encontrar os pontos comuns e posterior unir as cinco imagens através destes pontos comuns.

Este algoritmo é elaborado a partir do software de cálculo numérico Matlab e sua respectiva biblioteca para visão computacional, a Machine Vision Toolbox (MVT). Esta biblioteca fornece diversas ferramentas de visão computacional e controle baseado em visão, além de um completo guia para referência.

O trabalho está segmentado em cinco tópicos ao longo dos quais serão apresentadas uma breve contextualização, em seguida uma abordagem das metodologias utilizadas bem como a lógica de funcionamento, por fim são apresentados os resultados e fechamento é realizado com a conclusão.

## **2 CONTEXTUALIZAÇÃO**

Não é raro encontrar aplicações dentro do nicho de visão computacional onde é necessário corrigir ou relacionar imagens de um mesmo objeto, porém com perspectivas diferentes. Isto é, a imagem de um mesmo objeto, lugar ou pessoa é obtida de lugares distintos ou com câmeras calibradas de maneira divergente.

Dentre os exemplos destas aplicações podem ser citados a correção de perspectiva, correção de distorção, efeitos visuais (como Realidade Aumentada) e até mesmo na construção de panoramas, um dos objetivos deste trabalho.

A construção de panoramas é uma importante aplicação das ferramentas de visão computacional porque através dela são elaborados projetos arquitetônicos, mapas e rotas para localização e fotografias de objetos demasiadamente largos ou compridos.

É importante estudar e analisar as técnicas de visão computacional pois através dela são obtidas imagens com menos distorções e pontos impróprios, o que possibilita a criação de imagens com maior grau de fidelidade com a realidade.

### 3 METODOLOGIAS UTILIZADAS

Ao examinar o conjunto de imagens e verificar que é necessário construir uma única imagem fica evidente que:

- i. É preciso aplicar alguma técnica para encontrar os pontos correspondentes entre as imagens, saber qual a posição destes pontos e posteriormente unir os mesmos para que seja possível criar uma única imagem.
- ii. É preciso aplicar alguma técnica de correção de perspectiva, já que analisando as cinco imagens do conjunto é evidente cada imagem foi obtida a partir de uma configuração diferente da câmera, neste caso a cada imagem a câmera é rotacionada da esquerda para direita.

Atender o primeiro requisito (i) é possível através do algoritmo SURF (Speeded UP Robust Features), um detector e descritor de pontos de interesse que utiliza como base a matriz Hessiana. Tal algoritmo examina a imagem, após isto é capaz de extrair características como o e descrever possíveis pontos de interesse.

Através destes pontos de interesse é possível comparar duas imagens, ou seja, extrair pontos de uma imagem e comparar com pontos de uma segunda imagem. Caso estes pontos tenham características similares as imagens podem conter a mesma representação (mesmo objeto).

Baseado nesta funcionalidade, através do SURF é possível fazer a identificação de um objeto contido em uma imagem em comparação com outras imagens, o que é muito útil e atende o primeiro requisito. Além disso, o SURF ainda tem capacidade de fazer a comparação mesmo que existam rotações entre as imagens. A figura a seguir ilustra o conceito de comparação de pontos de interesse aplicado a primeira e segunda imagens:

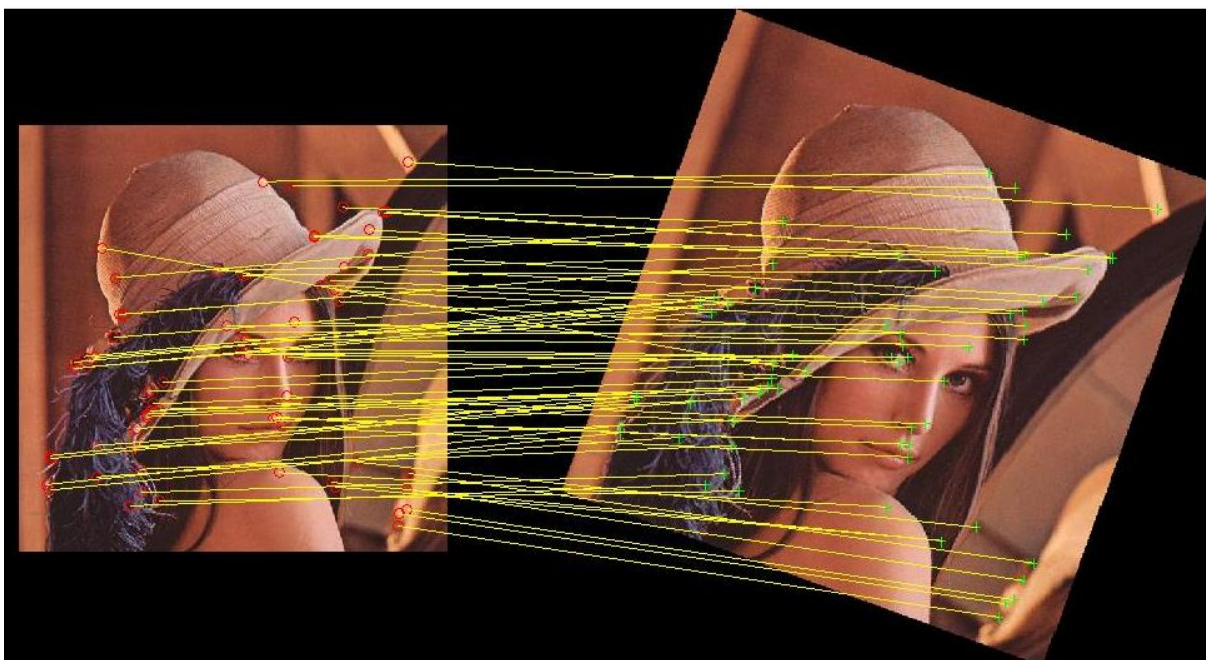


Fig.1 – Pontos de interesse comuns entre a primeira e a segunda imagem do conjunto – pontos obtidos com através das funções do Machine Vision Toolbox no MATLAB.

Em ambas as imagens são apresentados os pontos de interesse comuns, na primeira imagem são representados por círculos vermelhos enquanto na segunda imagem são representados por cruzeiros verdes.

Respeitar o segundo requisito (ii) é possível através do conceito de Homografia, que em síntese é uma transformação capaz de mapear pontos de um plano para outro plano, sendo que tais planos estão em orientações diferentes. Estas orientações diferentes neste caso são as rotações realizadas para tirar cada foto.

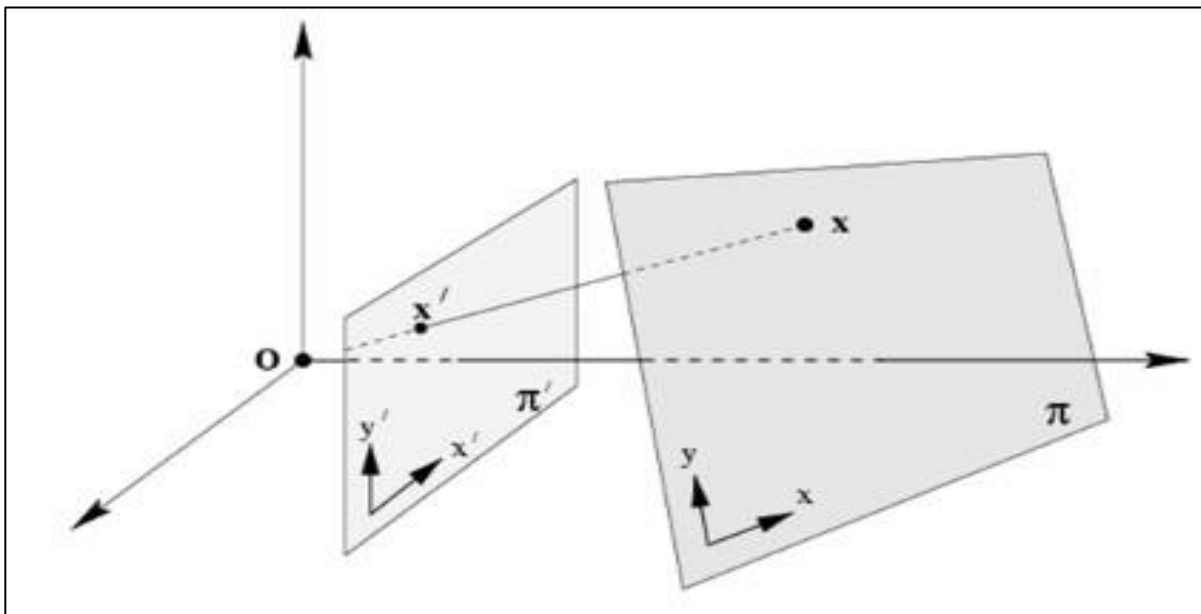


Fig.2 – Exemplo Homografia

Na imagem acima é ilustrado o conceito de homografia, o ponto  $x$  sobre o plano  $\pi$  é mapeado para o plano  $\pi'$  como um ponto  $x'$ , a transformação que relaciona os dois pontos é obtida através de uma matriz.



Fig.3 – Homografia da segunda imagem em relação a primeira imagem (a esquerda a imagem 2 original e a direita a imagem 2 após transformação) - resultado obtido com as funções do Machine Vision Toolbox no MATLAB.

Esta transformação da homografia relaciona a mudança entre dois planos, sendo assim é possível descobrir como ocorreu o deslocamento da câmera, e desta forma relacionar a configuração da primeira imagem com a configuração da segunda imagem, e assim sucessivamente para todas as imagens.

Com o conceito de Homografia e o algoritmo SURF as imagens podem ser unidas por meio dos pontos comuns, e em seguida uma imagem pode ser ajustada em relação a outra, e desta forma é possível montar um panorama a partir de várias imagens.

### 3 LÓGICA DE FUNCIONAMENTO

Antes de iniciar a construção do algoritmo foram realizadas pesquisas para encontrar soluções para problemas semelhantes, e assim utilizar tais soluções como base e referência para construir um algoritmo capaz de resolver o problema específico deste trabalho.

Após uma longa busca duas soluções parciais foram encontradas:

- Tópico “*Matching images with different orientations and scales in MATLAB*” do Stack Overflow.
- Exemplo da função “*matchfeatures*” disponível na seção de suporte do site da MathWorks.
- Exemplo “*Feature Based Panoramic Image Stitching*” disponível na seção de suporte do site da MathWorks.

Com estas soluções parciais e as funções do Machine Vision Toolbox do MATLAB o algoritmo de construção de panorama pode ser elaborado seguindo as seguintes etapas:

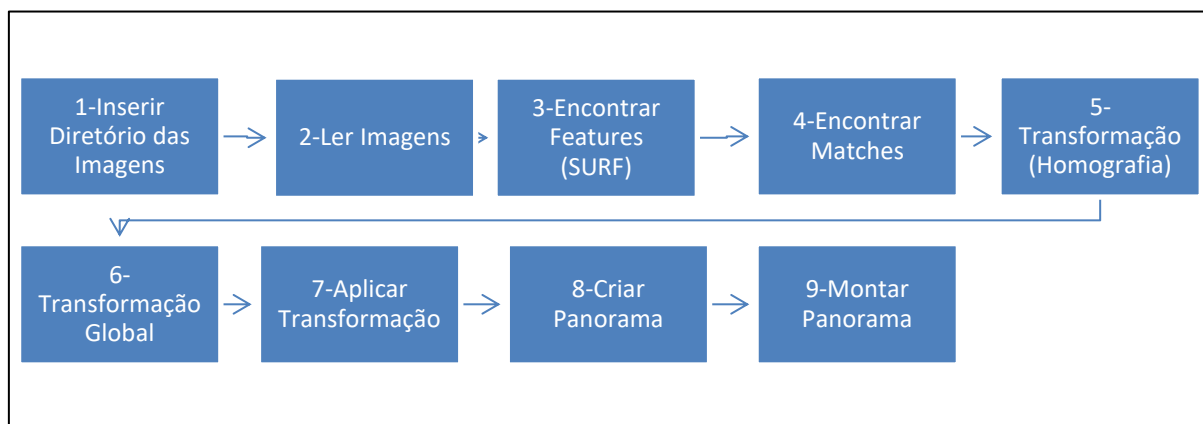


Fig.4 – Fluxograma do funcionamento do algoritmo.

As etapas que 3,4,5 e 6 devem ser repetidas para todos os pares de imagens, isto é, deve ser feito da primeira para a segunda imagem, da segunda para a terceira, da terceira para a quarta e assim sucessivamente até última imagem do conjunto. A seguir será uma breve explicação do código desenvolvido no MATLAB para realizar as etapas descritas acima tomando como exemplo o primeiro par de imagens (imagem 1 e imagem 2).

No MATLAB antes de iniciar o tratamento de imagem é necessário carregar (ler) as imagens. Para tanto, a função *imread* pode ser utilizada. O próximo passo é utilizar o algoritmo *SURF* para encontrar os pontos de interesse correspondentes entre as duas primeiras imagens.

Assim, em primeiro lugar a função *isurf* deve ser utilizada para encontrar pontos de interesse na imagem, tal função retorna um vetor de objetos do tipo *SurfPointFeature* que contém os pontos de interesse da imagem (como dito no tópico anterior, tais pontos independem de variações de escala ou rotação).

Com as features de cada imagem basta utilizar o método *match* do *SurfPointFeature* que retornará um objeto do tipo *FeatureMatch* que contém features comuns a ambas as imagens.





Fig.5 – Pontos de interesse comuns entre a primeira e a segunda imagem do conjunto – pontos obtidos com através das funções da biblioteca Vision do Matlab.

Devido a rotação da câmera da imagem 1 em relação a imagem 2 é necessário aplicar uma transformação para levar as duas imagens para a mesma perspectiva. Com de interesse comuns é possível realizar a transformação das imagens.

Tal transformação utiliza o conceito de homografia, que como enunciado anteriormente permite descobrir como ocorreu o deslocamento da câmera, e desta forma relacionar a configuração da primeira imagem com a configuração da segunda imagem.

Para a primeira imagem não existe uma imagem anterior, assim tanto a homografia quanto a transformação global serão neutras, o que quer dizer que sua matriz de transformação será uma identidade.

$$Trans\{1\} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Desta forma a primeira imagem será utilizada como referência, ou seja, todas as outras imagens serão ajustadas e calibradas com relação a primeira imagem, e como mostrado anteriormente a matriz da primeira transformação deve ser uma identidade.

Em sequência também deve-se definir homografia para a segunda imagem, que irá tratar da transformação da segunda imagem utilizando como referência a primeira imagem. No objeto *FeatureMatch* existe um método chamado *ransac* que pode calcular de forma iterativa a transformação a partir dos pontos de interesse comuns a ambas as imagens, neste caso os pontos encontrados anteriormente pelo SURF.

O *ransac* é um algoritmo capaz de ajustar e encontrar os melhores resultados para o um determinado modelo matemático, neste caso a Homografia. Após calcular a transformação que relaciona os pontos da segunda imagem para a primeira (Homografia), ainda é necessário determinar a transformação global que leva a imagem para a referência do panorama:

$$Trans(2) = Trans(2) * Trans(1) \quad (2)$$

Neste caso como a transformação é em relação a primeira imagem e a matriz *Trans(1)* é uma identidade os valores de *Trans(2)* não serão alterados.



Como dito anteriormente os procedimentos 3,4,5 e 6 devem ser repetidos para todos os pares de imagens do conjunto. Após calcular todas as homografias e transformações o passo seguinte é aplicar tais transformações nas respectivas imagens.



Fig.6 – Transformação da segunda imagem em relação a primeira (a esquerda a imagem 1, no centro a imagem 2 e na direita a imagem 2 após transformação) - resultado obtido com as funções do Machine Vision Toolbox no MATLAB.

Em seguida, com todas as imagens transformadas é possível montar o panorama, para tanto é necessário utilizar um tipo de filtro para adaptar cada imagem para a maneira correta e assim colar esta mesma imagem no panorama. Este tipo de filtro, também chamado de máscara, deve ter o mesmo formato da imagem transformada:

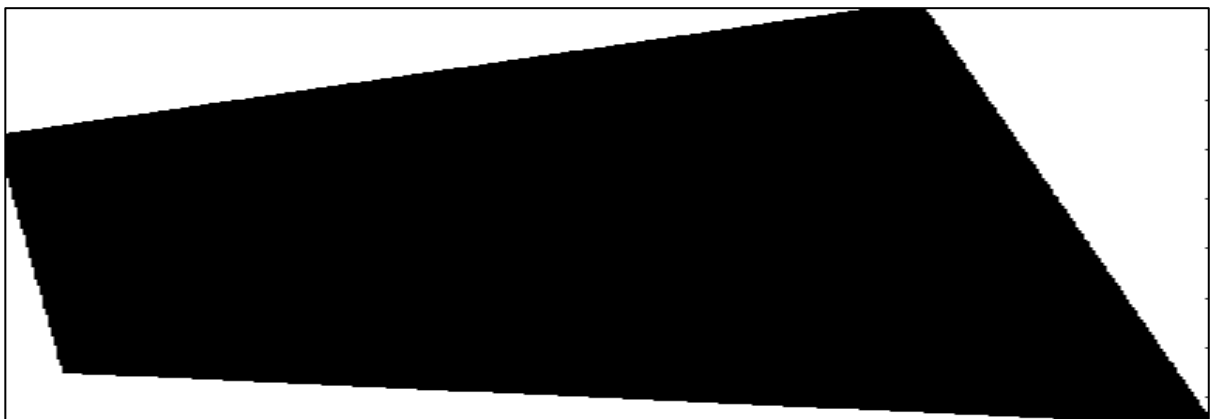


Fig.7 – Exemplo de máscara – resultado obtido com MATLAB.

Após criar uma máscara com as dimensões das imagens transformadas é preciso criar uma segunda máscara com as mesmas dimensões do panorama, e nesta segunda máscara será inserida máscara da respectiva imagem. Para inserir a máscara da imagem na máscara do panorama é necessário utilizar a função *ipaste* com a opção *add*.

Então basta realizar a multiplicação ponto a ponto da segunda máscara pelo panorama, isto fará com que o panorama deixe espaço livre com a dimensões adequadas para colar a imagem transformada. Finalmente com a função *ipaste* é possível inserir a imagem transformada no panorama.

### 3 TESTES E RESULTADOS

#### 3.1 Conjunto 1

O conjunto de imagens com trabalhado ao longo deste trabalho até o momento é apresentado a seguir:



Fig.8 – Conjunto de imagens 1

Após executar todas as operações em todos os pares de imagens basta unir as imagens transformadas e montar o panorama:

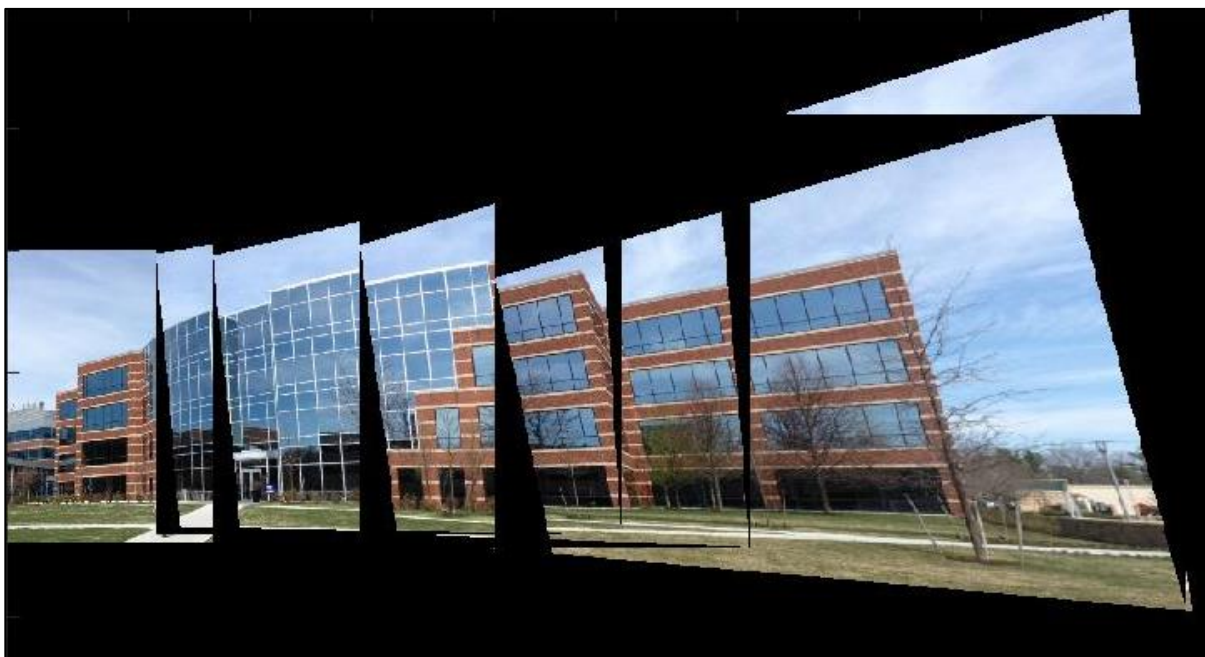


Fig.9 – Panorama - construído com a primeira imagem como referência. - resultado obtido com as funções do Machine Vision Toolbox no MATLAB.

Resolver este problema é possível após alterar da opção *extrapval* da função *homwarp* no momento de aplicar as transformações nas imagens. Então é obtido o seguinte resultado:



Fig.10 – Panorama ajustado - construído com a primeira imagem como referência - resultado obtido com as funções do Machine Vision Toolbox no MATLAB.

Ao analisar a figura 10 fica evidente uma divergência entre a parte inicial (esquerda) e a parte final (direita) do panorama, tal divergência pode ser classificada como uma deformação demasiada. Esta deformação é causada porque ao utilizar a primeira imagem como referência é necessário aplicar uma transformação aguda, isto porque a cada imagem a câmera é rotacionada, sendo assim a cada par de imagens o peso da transformação aumenta.

Para confirmar esta hipótese basta analisar a figura 10, a parte inicial que é composta pela primeira imagem sofreu pouca alteração (observar os contornos retos, pouca deformação e



tamanho próximo ao real), já a parte final composta pelas últimas imagens sofreu grandes alterações (observar contornos tortos, grande deformação e tamanho distante do real).

Em vista deste problema é proposta uma segunda abordagem para o problema do panorama, ao invés de utilizar a primeira imagem (imagem 1- conjunto 1) como referência utilizar a imagem central (imagem 3 – conjunto 1), desta maneira as transformações não serão tão agudas quanto no caso anterior.

Ao executar com a alteração na referência é obtido o seguinte resultado:



Fig.11 – Panorama construído com a terceira imagem como referência - resultado obtido com as funções do Machine Vision Toolbox no MATLAB.

É possível verificar que ao alterar a referência para a terceira imagem, a parte central sofreu pouca transformação (tamanho próximo ao real, contorno retos), o que era esperado. Ainda ao analisar a figura 8 fica evidente o acréscimo na qualidade do panorama, agora a deformação não está concentrada em uma área, pelo contrário, está espalhada nos dois cantos do panorama.

### **3.2 Conjunto 2**

Um segundo conjunto de imagens pode ser utilizado para confirmar os resultados obtidos anteriormente, desta forma o segundo conjunto de imagens é:



Fig.12 – Conjunto de imagens 2.

Utilizando como referência a imagem do extremo esquerdo é obtido o seguinte resultado:

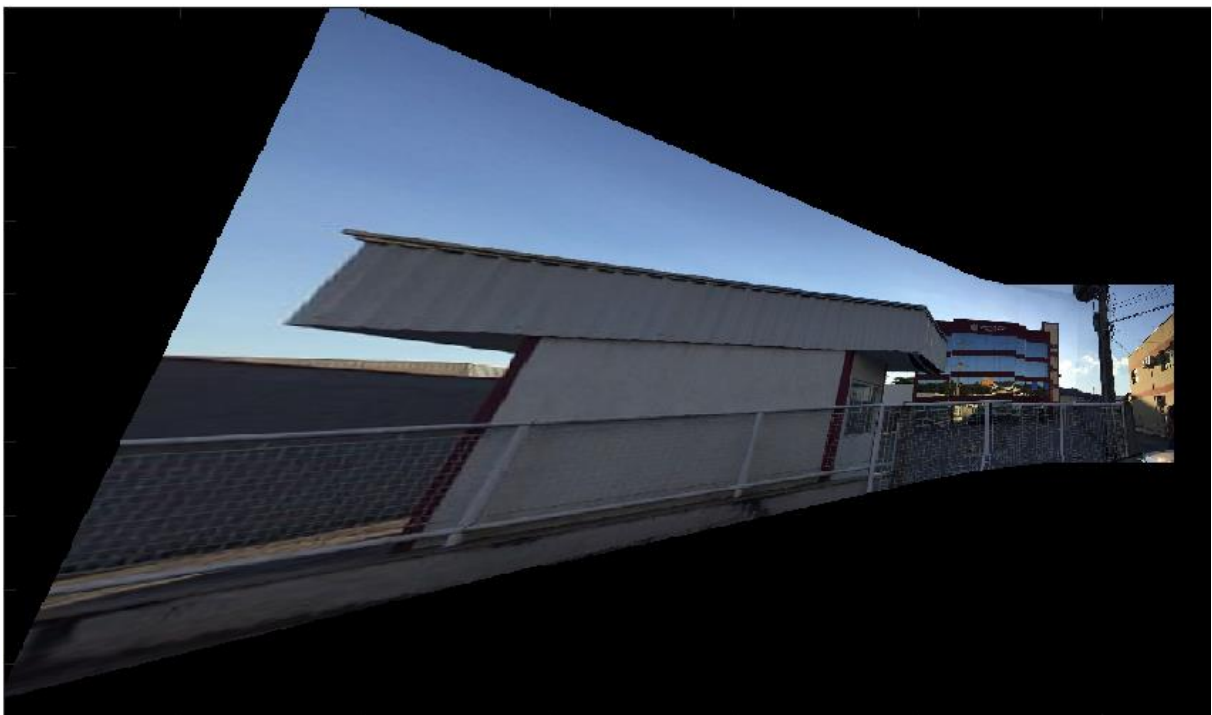


Fig.13 – Panorama construído com imagem do extremo esquerdo como referência - resultado obtido com as funções do Machine Vision Toolbox.

Novamente, em razão do grande deslocamento da câmera quando a imagem do canto é utilizada como referência ocorre uma deformação demasiada da imagem do canto contrário.

É possível aplicar a mesma metodologia anterior, e utilizar a imagem central como referência:



Fig.8 – Panorama construído com imagem do centro como referência - resultado obtido com as funções do Machine Vision Toolbox.

E assim como no conjunto 1, no conjunto de imagens 2 utilizar a imagem central como referência fornece um resultado muito melhor, ou seja, um panorama no qual é possível observar e analisar objetos com maior fidelidade ao mundo real.

## 7 CONCLUSÃO

Depois de concluir todas as etapas desde a modelagem ao projeto de controle e observação é possível relatar os pontos principais do trabalho. O primeiro ponto a ser avaliado é a eficiência do algoritmo, ou seja, através do resultado fica claro que o código desenvolvido é capaz de atingir o objetivo, que era construir um panorama.

Todos os conceitos abordados foram aplicados, verificados e confirmados através dos resultados, mais do que isso, foi possível verificar diferentes abordagens para resolver o mesmo problema.

O algoritmo é capaz de construir um panorama e utilizar como referência a imagem de um extremo (canto) ou a imagem central, o que proporciona a construção de um panorama de maior qualidade, isto é, com deformações menores. É importante ter este recurso, pois em alguns casos torna-se impraticável (grandes deformações) construir um panorama utilizando como referência a imagem de um dos extremos.

Outro ponto a ser avaliado é a maneira como o algoritmo é capaz de realizar as operações, em outras palavras o algoritmo desenvolvido pode ser aplicado a outros conjuntos de imagens para isto basta alterar a pasta onde estão armazenadas as imagens.

As grandes desvantagens encontradas ao longo da elaboração do trabalho foram a necessidade de ordenar o conjunto de imagens, colocar as imagens na ordem equivalente ao mundo real. Além disso, foram encontrados problemas com a definição do tamanho do panorama, para contornar alguns destes problemas a solução mais simples foi ajustar um panorama com grandes dimensões, o que pode ocupar espaço desnecessário na memória.

Por fim, ao reunir todas as características o algoritmo mostrou ser eficiente, porém não é capaz de atuar de maneira completamente automática.



## REFERÊNCIAS

BORTH, M. R. et al. Análise da Extração de Atributos do Algoritmo SURF em Espécies de Peixe. Seminário internacional de integração e desenvolvimento regional, [S.L], set. 2013. Disponível em: <[http://www.gpec.ucdb.br/pistori/publicacoes/borth\\_ecaeco2013.pdf](http://www.gpec.ucdb.br/pistori/publicacoes/borth_ecaeco2013.pdf)>. Acesso em: 18 abr. 2018.

OPENCV. Basic concepts of the homography explained with code. Disponível em: <[https://docs.opencv.org/3.4.1/d9/dab/tutorial\\_homography.html](https://docs.opencv.org/3.4.1/d9/dab/tutorial_homography.html)>. Acesso em: 11 abr. 2018.

PETER CORKE. Machine vision toolbox. Disponível em: <<http://petercorke.com/wordpress/toolboxes/machine-vision-toolbox>>. Acesso em: 09 abr. 2018.

SANTOS, Maikon Cismoski Dos. Revisão de Conceitos em Projeção, Homografia, Calibração de Câmera, Geometria Epipolar, Mapas de Profundidade e Varredura de Planos. Visão Computacional, Campinas, jun. 2012. Disponível em: <<http://www.ic.unicamp.br/~rocha/teaching/2012s1/mc949/aulas/additional-material-revision-of-concepts-homography-and-related-topics.pdf>>. Acesso em: 16 abr. 2018.

MONTANARI, Raphael. Detecção e classificação de objetos em imagens para rasteramento de veículos. São Carlos, 2015. Disponível em: <[http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08012016113715/publico/RaphaelMontanari\\_dissertacao\\_revisada.pdf](http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08012016113715/publico/RaphaelMontanari_dissertacao_revisada.pdf)>. Acesso em: 16 abr. 2018.