

# Reconstrução 3D a partir de imagens obtidas por uma aeronave não tripulada

Alexandre Yu Jin

Academia Militar, Lisboa, Portugal  
IST-UL, Lisboa, Portugal  
jin.ay@exercito.pt

Alexandre Bernardino

IST-UL, Lisboa, Portugal  
ISR, Lisboa, Portugal  
alex@isr.tecnico.ulisboa.pt

José Silvestre Silva

Academia Militar, Lisboa, Portugal  
LIBPhys-UC, Coimbra, Portugal  
jose.silva@academiamilitar.pt

**Resumo** — A criação de um modelo tridimensional altamente detalhado a partir de um terreno urbano pode estabelecer uma boa base para o planeamento de uma operação militar. A partir dela é possível obter informação espacial adicionada sobre o espaço em questão, que, em inúmeros casos, constitui um local remoto com difíceis condições de acessibilidade. A utilização de veículos aéreos não tripulados pode mitigar este problema, a sua capacidade de realizar voos de baixa altitude permite para o operador obter imagens de forma mais económica sem a necessidade de se preocupar problemas relacionados com a gestão e a segurança dos recursos humanos e dos meios disponíveis.

No presente artigo, três *datasets* de imagens captadas por um UAV em contexto operacional, fornecidos pela Guarda Nacional Republicana, serão usados. O principal objetivo será desenvolver as reconstruções tridimensionais das respetivas áreas cobertas com base numa técnica chamada *Structure from Motion*. Neste artigo também será realizado um estudo exaustivo de diferentes algoritmos de extração de *features*, SIFT, HAHOG, ORB<sub>(10000)</sub> e AKAZE. Para a correspondência entre *features* é usado uma procura FLANN e para a rejeição de *outliers* é usado uma *Random Sample Consensus*. Os resultados, serão apresentados tanto quantitativamente, através de quantidade procura e correspondência de *features* e os seus tempos respetivos, tempo das reconstruções incrementais e tempo total do algoritmo de reconstrução em cada *dataset*. Os algoritmos SIFT e HAHOG foram os que obtiveram os melhores resultados, enquanto ORB<sub>(1000)</sub> foi o método menos eficiente tanto quantitativamente como na qualidade das respetivas reconstruções.

Palavras-chave: UAV; Reconstrução tridimensional, Structure-from-Motion; Extração de *features*; SIFT, HAHOG, ORB, AKAZE.

## I. INTRODUÇÃO

As tecnologias por detrás de UAVs sofreram um prodígio crescimento na última década, desde a simples pilotagem remota até as mais recentes atualizações de controlo autónomo e a capacidade de seguir uma rota pré-definida. Embora o equipamento responsável pela captura de imagens em *drones* comerciais não seja tão técnico como Scanners de laser terrestres (TLS), ou sensores laser LiDAR para a modulação 3D, continuam ainda a trazer inúmeras vantagens para aplicações no mundo real. Do seu tamanho viável, relativamente barato e de

fácil manuseamento, a principal vantagem dos UAVs é a sua capacidade de aquisição de dados em diferentes ângulos de perspetiva aérea.

A reconstrução de modelos 3D a partir de imagens aéreas tem sido um dos tópicos cruciais na investigação da Visão Computacional e Fotogrametria Digital, uma vez que oferece uma distribuição espacial detalhada relacionada com a área ou objeto de interesse. Este tipo de reconstrução é usado em inúmeras áreas, tais como a construção civil, turismo, transportes e a militar. Embora os recentes progressos significativos nesta área, continua a não existir um algoritmo autónomo de construção 3D que consegue cobrir todos os domínios de uma só vez. Várias técnicas têm sido propostas na literatura para gerar automaticamente estes modelos. Algumas visam obter a maior precisão possível com a utilização de LiDARs, Laser Scanners, outros estudos procuram a utilização do Sistema de Navegação Inercial e o Sistema de Posicionamento Global como instrumentos auxiliares para alcançar melhores precisões.

Structure-from-Motion (SfM) é uma técnica usada na Visão Computacional que visa a reconstrução de uma estrutura de uma cena tridimensional. A reconstrução 3D é baseada num conjunto de imagens 2D com um elevado grau de sobreposição. SfM envolve inúmeros passos a serem seguidos, incluindo, a deteção de *features* (também conhecidas por *keypoints* ou *interest-points*), a sua correspondência, cálculo do movimento da câmara e a estimação de poses. Teoricamente, é necessário que o processo de deteção de *features* seja robusto e que consiga de forma consistente detetar os mesmos pontos dentro de um conjunto com múltiplas imagens, independentemente das suas rotações, escalas ou iluminação. As *features* após extraídas, são caracterizadas dependendo do método proposto, são comparadas em relação as suas imagens de origem, criando uma estimativa de matriz de calibração da câmara, da matriz de calibração e uma previsão do movimento da câmara. Consoante o algoritmo usado as *keypoints*, podem ser detetadas em várias formas, cantos, pontos circulares, junções, linhas, *etc.* Após as suas extrações, elas são descritas logicamente consoante os padrões encontrados nas suas vizinhanças, este processo é chamado como *feature description*, visto que para cada *feature*, é atribuída uma identidade para que esta possa ser comparada na fase de correspondência.

## II. ENQUADRAMENTO TEÓRICO

SfM é uma técnica de *low-cost* de fotogrametria para a reconstrução topográfica a altas resoluções. É especialmente usada no estudo de área remotas com difícil acesso. SfM segue os mesmos princípios que a fotogrametria estereoscópica, nomeadamente que uma reconstrução tridimensional pode ser realizada usando múltiplas imagens em diferentes perspectivas de visualização desde que exista um elevado grau de sobreposição entre elas, como se pode verificar na Figura 1[1].

De acordo com a literatura [1], o primeiro princípio da SfM, é a obtenção da localização tridimensional de um ponto num determinado cenário. Ao contrário de métodos tradicionais de fotogrametria que necessitam de informação *a priori* sobre os parâmetros da câmara e a localização do cenário, usando as técnicas de SfM, não é necessário a aquisição deste conhecimento, visto que estas são obtidas no ato da aquisição das poses das câmaras e da geometria dos cenários durante o processo da correspondência de *features*. Visto que as posições das câmaras são calculadas através das imagens, a escala e a orientação poderão ter erros associados. Estes parâmetros que estão de acordo com uma perspectiva imagem-espaco, onde deveriam estar alinhadas de acordo com um sistema de coordenadas objeto-espaco. Como solução, pontos com coordenadas objeto-espaco que se sejam conhecidos dentro das amostras de imagens devem ser inseridos. A seleção destes pontos *Ground-Control*, pode ser simplificada colocando um alvo físico definido por um centroide no cenário, com um elevado contraste, antes da aquisição as imagens.

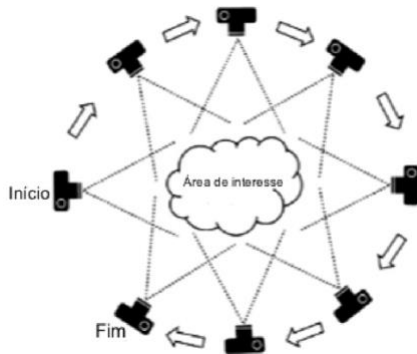


Figura 1. Captura de um conjunto de imagens com sobreposição. Adaptado de [1]

Em poucas palavras, um algoritmo de SfM é constituído por dois módulos, Procura de Correspondências e a Reconstrução Incremental como é possível observar na Figura 2. Após a aquisição das imagens a serem usadas, uma extração de *features* é realizada, de seguida é feita uma procura de pares entre *features* nas diferentes imagens e por fim uma verificação geométrica. Para a reconstrução incremental, um par de imagens é usado para a sua inicialização, as imagens são então registadas para a reconstrução 3D, sofrendo também um processo de triagulação e de *bundle adjustment*. Para a eliminação de pontos falsos é realizado um filtramento de *outliers* [2].

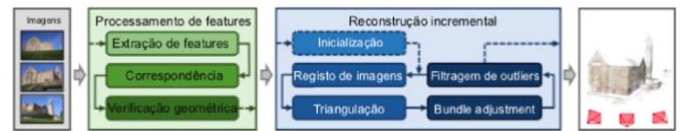


Figura 2. Funcionamento de um algoritmo SfM. Adaptado de [2]

### A. Procura de Correspondências

O módulo da procura de correspondências tem como papel de identificar os pontos projetados nas imagens que estejam sobrepostos. Possui três passos sequenciais: extração de *features*, a sua correspondência e a verificação geométrica.

O primeiro passo, extrai um conjunto de pontos para cada imagem. Cada ponto, corresponde a um conjunto de informação, contendo a respetiva localização e a sua descrição. Existem inúmeros algoritmos usados para este problema como *Harris Corner Detection* [3], *Kanade-Lucas-Tomasi* [4], *Scale-Invariant Feature Transform (SIFT)* [5], entre outros. Uma *feature* em si pode ser um simples ponto, canto de uma estrutura, ou até mesmo um objeto, no entanto a sua seleção deve seguir as seguintes propriedades [6]:

- **Repetibilidade:** Podendo ser considerado a propriedade mais importante, havendo um par de imagens que foram captadas em diferentes perspectivas de visualização do mesmo cenário, uma elevada percentagem dos mesmos pontos devem estar presentes. Para isso dois atributos são necessários, invariância e robustez.
- **Distintividade:** Uma elevada variação na intensidade de padrões dentro do conjunto de *features* deve existir, de modo que estas possam ser encontradas e correspondidas.
- **Localização:** Permite tanto que a chance de obstrução seja reduzida como para um modelo mais simples de aproximação geométrica e de deformações fotométricas entre duas imagens.
- **Quantidade:** O número ótimo de *features*, varia de situação para situação, no entanto, a densidade das mesmas deve refletir a informação contida dentro de uma imagem oferecendo uma representação compacta.
- **Exatidão:** A localização precisa dos pontos deve existir, para que no ato de comparação não existam problemas tanto em termos de escalabilidade como em forma das mesmas.
- **Eficiência:** O tempo de deteção das *features* numa imagem nova deve ser tomada em consideração dependendo da aplicação da reconstrução.

Durante passo de correspondência, pares de imagens são combinadas de acordo com a presença de cada *feature*. Num conjunto de  $N$  imagens,  $\frac{N(N-1)}{2}$  pares podem existir. Este processo pode ser exaustivo e dispendioso se uma abordagem ingénua de comparar todas as imagens uma a uma é usada. A complexidade é equivalente a  $O(N^2F^2)$ , sendo  $F$ , o máximo de *features* presente nas imagens [6]. Para este problema de escalabilidade, uma otimização deve ser usada, tal como uma procura de *approximation nearest neighbor kd-tree*.

Durante o passo anterior, pares de imagens são obtidos baseando nas suas presenças, logo, não existe uma garantia que as respetivas *features* representem o mesmo ponto no cenário, para isso, é necessário a verificação geométrica. Existem

inúmeras técnicas que permitem descrever uma relação geométrica entre duas diferentes perspetivas. Uma homografia pode ser usada para revelar uma transformação entre duas imagens. Se os parâmetros intrínsecos da câmara são conhecidos, uma abordagem epipolar geométrica permite obter o movimento da câmara através de uma matriz essencial. No final, para remover pontos falsos encontrados neste passo, uma técnica de filtramento chamada RANSAC [7] é usada.

**B. Reconstrução Incremental**

Como é possível observar a reconstrução incremental possui cinco passos, a inicialização, registo das imagens, triangulação, *bundle adjustment*, e por fim a eliminação de *outliers*. A inicialização consiste na escolha de um primeiro par de imagens correspondidas. Esta escolha deve ser cuidada, visto que o resto da reconstrução depende dela, e que esta poderá não recuperar de uma má inicialização. Idealmente o modelo deve iniciar numa localização densa onde as imagens em questão tenham o maior degrau de sobreposição. Desta maneira, é possível ter uma melhor robustez e precisão na reconstrução devido a uma elevada redundância. No caso de uma má inicialização, na realização do *bundle adjustment*, não existirá informação suficiente para refinar as posições dos pontos e das poses das câmaras [8].

Durante o registo de imagens, novas imagens são adicionadas à reconstrução e identificadas como imagens registadas. Para cada imagem registada, a pose da câmara  $P$  é calculada.  $P$  tem que ser definido numa relação tridimensional de acordo com o sistema de coordenadas global e de rotação (eixo vertical, eixo de longitude e eixo de latitude). Esta informação pode ser obtida através de um problema *Perspective-n-Point*.

A triangulação, ilustrada na Figura 3, é um passo essencial, consiste na estimação da localização de um ponto 3D que é observado em múltiplos ângulos. A estabilidade do modelo aumenta consoante a redundância, através da expansão da cobertura do cenário definido por um conjunto de pontos  $X$ . Quando um novo cenário de pontos se encontra presente em diferentes visualizações de duas ou mais imagens, o respetivo novo conjunto de pontos,  $X_k$ , são registados e adicionados para o conjunto  $X$  [9].

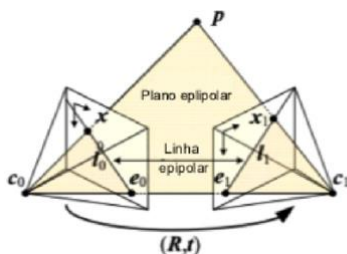


Figura 3. Triangulação tridimensional para a obtenção do ponto P. Adaptado de [9]

Por fim, é realizado o *bundle adjustment*, este, constituído o processo de otimização e refinação do modelo tridimensional e da estimação das poses da câmara, minimizando o erro de reprojeção entre diferentes poses,  $P$ , e dos parâmetros do ponto  $X_k$ .

**III. METODOLOGIA DE INVESTIGAÇÃO**

A proposta de implementação encontra-se ilustrada na Figura 4. É constituída por dois módulos principais, *Structure-from-Motion* constituído pelos dois sub-módulos já explicados anteriormente e a o processamento da nuvem de pontos.

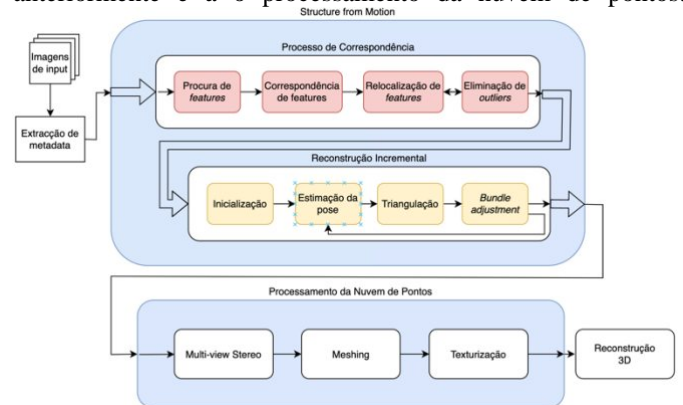


Figura 4. Proposta de fluxo de trabalho do algoritmo

**A. Processo de Correspondência**

A primeira etapa será responsável pela extração de *features* para cada imagem. Quatro algoritmos são testados, SIFT, *Accelerated KAZE* (AKAZE) [10], *Oriented FAST and Rotated BRIEF* (ORB) [11], e *Hessian Affine and Histogram of Gradients* (HAHOG) [12].

Na segunda etapa, o algoritmo *Fast Library Approximation Nearest Neighbors* (FLANN) é aplicado para cada par de imagens dos descritores das *features*, com a exceção quando se o método ORB que não é compatível, visto que usa um descritor binário, como alternativa uma abordagem de força bruta é usada. Para evitar pares falsos é usado um limite de distância que é usado com o rácio de teste. Na presença de duas imagens,  $X$  e  $Y$ , o descritor usado em  $X$  satisfaz-se com dois pontos vizinhos mais próximos de  $Y$ , com respetivas distâncias,  $d_1$  e  $d_2$ , caso  $\frac{d_1}{d_2} < 0,6$ .

A última etapa é constituída pela verificação geométrica e a remoção de *outliers*. Um algoritmo RANSAC é usado com um limite de 0,6% da dimensão máxima da imagem para a estimar a matriz fundamental. Um algoritmo de oito pontos é usado a cada iteração para determinar a normalização da matriz fundamental candidata. A matriz retornada pelo RANSAC é refinada usando um algoritmo *Levenberg-Marquardt* para os oitos pontos. Para os pontos na matriz recuperada que não seguem o limite de 0,6 são removidos.

**B. Reconstrução Incremental**

Dos inúmeros tipos de SfM presentes na literatura, a que foi implementada, foi a incremental, viste que é mais completo e oferece níveis maiores de precisão e robustez. Assim sendo esta etapa é constituída por três níveis:

- **Inicialização:** Tem objetivo de procurar pelo primeiro par de imagem a serem registadas. Este par deve ter um nível significativo de paralaxe, isto significa que a posição da câmara entre duas capturas de imagens, deve ter uma diferença grande o suficiente em relação à distância para o cenário. Para encontrar o nível de

paralaxe, é realizada um ajuste do modelo rotacional da câmara. Somente pares em que neste modelo não é possível a sua explicação, ou seja, com um valor superior a 30% de *outliers*, são considerados. Estes pares são então listados e ordenados de acordo o grau de *outliers* encontrados.

- **Registo do primeiro par de imagens:** O primeiro par na lista criada é testado. Caso a inicialização falhe, o próximo índice é testado. Caso aceite, é registado, e o processo de aumentar o tamanho através de novas imagens na construção é iniciada. A reconstrução do primeiro par baseia-se num algoritmo de cinco pontos. Se existirem *inliers* suficientes, este par inicial sofre com o processo de triangulação e de *bundle adjustment*
- **Registo individual de imagens:** Neste passo final, são adicionadas novas imagens para a reconstrução. As fotos ainda não incluídas, que possuem pontos já adicionados à reconstrução, são prioritários. A estimação da pose, é aplicada de maneira que cada imagem fique bem alinhada com o modelo existente. Na existência de novos pontos, estes são triangulados com os que já foram adicionados. Caso seja necessário, ao modelo é aplicado o *bundle adjustment* e re-triangulado. O último passo consiste na integração das posições GPS, obtida pela extração de *metadata*.

### C. Point Cloud Handling

Com a nuvem de pontos esparsa criada no passo anterior, a função deste novo bloco de trabalho é adicionar textura à nuvem de pontos. Uma biblioteca *open-source* chamada OpenMVS [14] é usada. Este bloco, possui três componentes: *Multi-view Stereo*, que procura aumentar a precisão da nuvem de pontos desenvolvendo mapas de profundidade, *meshing*, que produz uma malha de textura para a nuvem de pontos e a texturização, que aplica a malha de textura na nuvem pontos, criando assim uma reconstrução texturizada.

## IV. SIMULAÇÃO E DISCUÇÃO DE DADOS

Para as simulações realizadas três *datasets*, fornecidas pela Guarda Nacional Republicana, foram utilizadas. Cada amostra corresponde a diferentes cenários operacionais que ocorreram. As principais características para cada *dataset* são:

- O primeiro conjunto possui 108 imagens. É caracterizada como um local urbano com uma rua principal, e algumas moradias ao longo da sua extensão. A trajetória realizada pelo *drone* tem uma forma retangular com *zig-zags* por dentro do retângulo.
- O segundo *dataset*, apesar de representar uma localidade diferente, possui propriedades semelhantes em relação ao tipo de cenário e à trajetória. A maior diferença encontra-se no cenário presente no meio do cenário, representando um acidente, que originou esta operação. Este conjunto possui 46 imagens. O principal objetivo será avaliar o funcionamento do algoritmo caso a número de amostras presentes seja de um número reduzido.

- O *dataset* final que possui 72 imagens, representa um local mais rural, mais objetos são considerados, nomeadamente, uma estrada principal curva, mais vegetação, pessoas e moradias. Um trajeto circular foi adicionado ao *drone*. Assim o objetivo será avaliar o aumento de *performance* quando mais perspetivas de visualização são adicionadas.

### A. Primeiro dataset

O primeiro conjunto de imagens representam um local em Vila Real, Portugal. Observando a Tabela 1, que representa uma avaliação quantitativa, é possível observar informação relativa ao número de *features* extraídas e reconstruídas, o tempo alocado para cada módulo na construção da nuvem de pontos e o tempo total usado para a reconstrução tridimensional. É possível observar que SIFT foi o método que encontrou mais *features* por imagens, no entanto o método HAHOG, foi mais eficiente conseguindo mais *features* reconstruídas. Comparando os tempos de cada método, ORB conseguiu encontrar os pontos muito mais rapidamente, no entanto, no ato da correspondência entre as mesmas, foi a que demorou mais, devido à sua incompatibilidade com o algoritmo *FLANN*. *AKAZE* por outro lado foi o método que teve a reconstrução mais rápida com 13 minutos e 32 segundos.

TABELA 1 - ANÁLISE QUANTITATIVA EM RELAÇÃO COM AS FEATURES OBTIDAS DURANTE O BLOCO DE STRUCTURE FROM MOTION PARA O DATASET Nº1

Dataset 1	Features detetadas			Features reconstruídas (matches - outliers detected)			Tempo alocado para a reconstrução da nuvem de pontos			Tempo de reconstrução do algoritmo
	Min	Max	Média	Min	Max	Média	Procura	Correspondência	Reconstrução incremental	
SIFT	10026	37324	15043	110	8827	3586	1'38"	1'44"	1'40"	17'21"
AKAZE	1749	9998	6176	38	3104	1143	0'39"	0'30"	0'34"	13'32"
ORB	9850	10000	9994	52	3562	1002	0'11"	9'53"	0'33"	20'26"
HAHOG	12248	17120	13464	100	8791	4562	0'49"	1'21"	2'4"	16'30"

À medida que as nuvens de pontos são criadas, falta ao algoritmo a realização da reconstrução da nuvem de pontos texturizada geo-referenciada, ilustrada na Figura 5. Apenas a nuvem de pontos do *SIFT* é mostrada, uma vez que foi a que atingiu uma das duas melhores performances, os outros modelos de texturas de extracção de métodos podem ser considerados como não significantes. Desde as nuvens de pontos até à texturização, muita coisa melhorou. A *pointcloud* criada, tornou-se agora muito mais legível, e quase tudo pode ser identificado. As áreas brancas apresentadas anteriormente foram preenchidas, embora sofrendo ainda algum ruído. As casas tornaram-se muito claras, minimizando algumas das regiões brancas abaixo dos telhados, e finalmente, alguma percentagem de vegetação é agora mais visível.



Figura 5. Reconstrução texturizada vista de dois lados opostos, usando o método SIFT no *dataset* nº1

B. Segundo dataset

Sendo que o *dataset 1* tinha um número decente de imagens de *input*, no segundo conjunto de dados, elas são cortadas ao meio. Os elementos encontrados são muito semelhantes, uma estrada e algumas casas nas laterais, mas a área coberta é menor.

A informação relacionada com a extracção dos elementos é representada na Tabela 2. Uma correlação significativa esperada com o primeiro conjunto de dados é encontrada, uma vez que cada método de extracção realizado é o mesmo, e o cenário também semelhante, verificando-se apenas uma diferença significativa no número de imagens. *SIFT* foi o método que extraiu as mais *features* médias por imagem, 14200, mas foi *HAHOG* que atingiu o maior par de características construídas. *AKAZE* foi o mais rápido, terminando a reconstrução, em cinco minutos e cinquenta e um segundos. Ao mesmo tempo, *ORB* foi o mais lento, oito minutos e quarenta e dois segundos, pela mesma razão.

TABELA 2 - ANÁLISE QUANTITATIVA EM RELAÇÃO COM AS FEATURES OBTIDAS DURANTE O BLOCO DE STRUCTURE FROM MOTION PARA O DATASET Nº2

Dataset 2	Features detetadas			Features reconstruídas (matches - outliers detected)			Tempo alocado para a reconstrução da nuvem de pontos			Tempo de reconstrução do algoritmo
	Min	Max	Média	Min	Min	Max	Média	Matching	Min	
SIFT	10068	22083	14200	1099	10776	4650	0'31"	0'32"	0'42"	7'36"
AKAZE	2459	9997	7248	517	4076	1785	0'13	0'12	0'23	5'51"
ORB	9466	10000	9968	221	3857	1546	0'05"	3'55"	0'29"	8'42"
HAHOG	12152	15498	14083	1530	10025	5531	0'19"	0'29"	0'48"	7'33"

Uma vez que, quantitativamente, *AKAZE* foi o método mais rápido, reconstruindo 1785 correspondências, pode concluir-se que, neste caso, este algoritmo de extração foi mais eficiente em comparação com os outros três conjuntos de dados.

A fase final do algoritmo encontra-se na Figura 6. A reconstrução texturizada parece ter a maior parte dos espaços em branco complementados. De maneira geral, a texturização obtida a partir das imagens de entrada parece ter funcionado muito bem. As fachadas das casas são agora preenchidas corretamente, exceto para uma pequena área abaixo dos respetivos telhados.

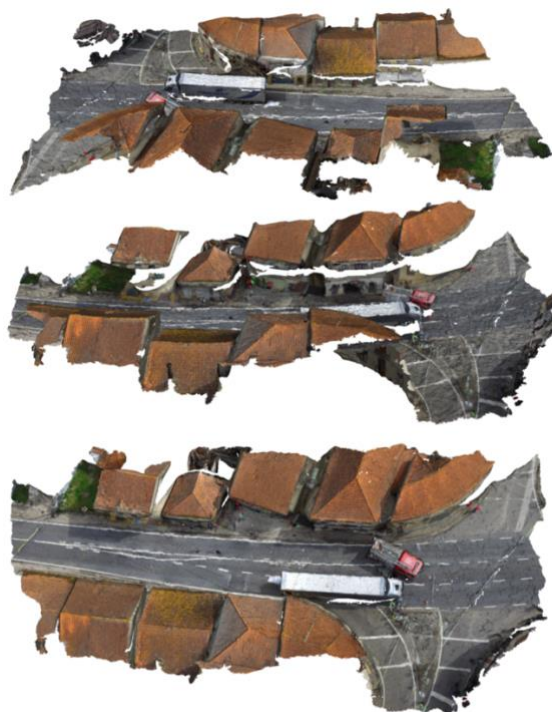


Figura 6. Reconstrução texturizada vista de dois lados opostos e uma visualização oblíqua, usando o método AKAZE no dataset nº2

C. Terceiro dataset

Analisando a informação fornecida na Tabela 3, em contraste com os testes anteriores, *SIFT* foi o método capaz de extrair o maior número de características, 17790 por imagem, sendo também aquele com a percentagem mais considerável de correspondências reconstruídos. Como esperado, o tempo de reconstrução total *ORB* permaneceu elevado aos 26 minutos e 20 segundos, e a sua qualidade de detecção de características voltou a ser baixa, uma vez que a percentagem de pares de *features* construídos também é fraca devido à sua ineficiência em problemas de escalabilidade.

TABELA 3 - ANÁLISE QUANTITATIVA EM RELAÇÃO COM AS FEATURES OBTIDAS DURANTE O BLOCO DE STRUCTURE FROM MOTION PARA O DATASET Nº3

Dataset 3	Features detetadas			Features reconstruídas (matches - outliers detected)			Tempo alocado para a reconstrução da nuvem de pontos			Tempo de reconstrução do algoritmo
	Min	Max	Média	Min	Max	Média	Procura	Correspondência	Reconstrução incremental	
SIFT	10255	32319	17790	1812	17151	8072	0'40"	1'04"	3'50"	22'05"
AKAZE	8460	9997	9867	629	6071	3666	0'28"	0'28"	2'01"	20'20"
ORB	10000	10000	10000	380	4696	2732	0'08"	5'29"	0'49"	26'20"
HAHOG	12645	16154	13416	2057	9251	6658	0'29"	0'42"	1'22"	21'24"

A reconstrução texturizada referenciada que se encontra na Figura 7 parece melhorar a partir das nuvens de pontos criadas. A informação retirada das imagens tornou-se agora texturizada com sucesso nas construções da implementação. Os objectos dentro das casas e dos carros, tais como janelas, portas e cores, podem agora ser bem percebidos e verificados. A principal deficiência encontrada encontra-se na parte de trás da casa, uma vez que algumas regiões não puderam ser texturizadas uma vez

que os disparos dos drones são demasiado verticais, resultando numa falta de informação sobre essa parte.



Figura 7. Reconstrução texturizada vista de dois lados opostos e uma visualização oblíqua, usando o método SIFT no *dataset* n.º3

## V. CONCLUSÕES

Como facilmente se pode concluir, podemos afirmar que o conjunto de dados n.º3 alcançou os melhores resultados. Três conjuntos de dados foram testados em condições diferentes para avaliar o desempenho da implementação. O primeiro conjunto de dados possuía muitas imagens de entrada com uma determinada trajetória. No segundo *dataset* a trajetória era semelhante, no entanto o número de imagens tiradas foi reduzido em aproximadamente metade. No último conjunto de dados, foi adicionada uma trajetória circular em torno do cenário. Em primeiro lugar, como as variáveis testadas têm uma importância diferente, não se deve depender inteiramente de uma. Em segundo lugar, verificou-se que embora seja recomendado um número mais elevado de imagens, não trouxe um valor de grande impacto às nuvens de pontos, uma vez que a informação respetiva que acrescenta pode não significar muito se imagens com elevada redundância estiverem presentes. Em terceiro lugar, a importância de uma trajetória bem planeada com um ângulo de inclinação que cubra o suficiente, quer de tetos quer das laterais de um objeto, é muito mais influente do que o número de imagens, uma vez que traz muito mais informação de qualidade conduzindo a melhores pontos-chave e aos seus descritores. Finalmente, adicionar uma textura de

georreferência ao modelo pode dar o toque final de saída necessário para uma reconstrução melhorada.

## REFERÊNCIAS BIBLIOGRÁFICA

- [1] M. Westoby, J. Brasington, N. Glasser, M. Hambrey, and J. Reynolds, 'structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications, *Geomorphology*, vol. 179, pp. 300–314, 2012.
- [2] J. L. Schonberger and J.M. Frahm, *Structure-from-motion revisited*, *Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 4104–4113.1, 2016.
- [3] C. Harris and M. Stephens, *A Combined Corner and Edge Detector*. *Proceedings of the 4th Alvey Vision Conference*. Vol. 15. 1988
- [4] B. Lucas and T. Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision*, *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [5] D. Lowe, *Object recognition from local scale-invariant features*, *Proceedings of the International Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999.
- [6] T. Tuytelaars and K. Mikolajczyk, *Local invariant feature detectors: A survey*, *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.
- [7] H. Cantzler, *Random sample consensus (ransac)*, *Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh*, 1981.
- [8] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski, *Building rome in a day*, *2009 IEEE 12th International Conference on Computer Vision*, pp. 72–79, 2009.
- [9] R. Szeliski, *Structure from motion*, *Springer London*, pp. 303–334, 2011.
- [10] D. Lowe, *Object recognition from local scale-invariant features*, in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 vol.2, 1999
- [11] P. Fernández Alcantarilla, *Fast explicit diffusion for accelerated features in nonlinear scale spaces*, 09 2013.
- [12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, *Orb: An efficient alternative to sift or surf*, in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.
- [13] D. Cernea, *OpenMVS: Multi-view stereo reconstruction library*, <https://cdcseacave.github.io/openMVS>, 2022,