



LABORATÓRIO DE ÓPTICA E MECÂNICA EXPERIMENTAL

RELATÓRIO INTERNO

***Reconstrução e Caracterização de
Estruturas Anatômicas Exteriores
usando Visão Activa***

Teresa Cristina de Sousa Azevedo

João Manuel R. S. Tavares

Mário A. P. Vaz

Setembro de 2005



Universidade do Porto

Faculdade de Engenharia

FEUP

 **inegi**
Instituto de Engenharia
Mecânica e Gestão Industrial

Reconstrução e Caracterização de Estruturas Anatómicas Exteriores usando Visão Activa

Por:

Teresa Cristina de Sousa Azevedo

Licenciada em Engenharia Electrotécnica e de Computadores pela
Faculdade de Engenharia da Universidade do Porto (2002)

Orientador:

João Manuel R. S. Tavares

Professor Auxiliar da Faculdade de Engenharia da Universidade do
Porto, Departamento de Engenharia Mecânica e Gestão Industrial

Co-orientador:

Mário A. P. Vaz

Professor Associado da Faculdade de Engenharia da Universidade do
Porto, Departamento de Engenharia Mecânica e Gestão Industrial

RESUMO

Este trabalho teve como principais objectivos a familiarização e análise de técnicas de Visão Computacional para a Reconstrução Tridimensional de Objectos, tendo sido iniciado o desenvolvimento de uma plataforma computacional.

O presente relatório é constituído por três capítulos: no primeiro, são definidos os objectivos; no segundo, é apresentada a plataforma computacional, assim como alguns resultados experimentais; no terceiro e último capítulo, são apresentadas as conclusões relativas ao estudo e ao desenvolvimento realizado e, finalmente, são referidas as perspectivas de trabalho futuro.

ABSTRACT

The work carried through had as main goals the familiarization and analysis of Computer Vision techniques for Three-Dimensional Reconstruction of Objects. From this work the development of a computer platform has been initiated.

The present report has three chapters: in the first one, the goals are defined; in the second, the computer platform is presented, as well as some experimental results; in the third and last chapter, the conclusions relative to the study and work done are drawn and, finally, some perspectives of future work are given.

Índice

CAPÍTULO 1.	INTRODUÇÃO	1
1.1.	INTRODUÇÃO À VISÃO COMPUTACIONAL.....	2
1.2.	RECONSTRUÇÃO 3D DO CORPO HUMANO	3
1.3.	OBJECTIVOS	4
CAPÍTULO 2.	PLATAFORMA COMPUTACIONAL.....	7
2.1.	PLATAFORMA COMPUTACIONAL EM DESENVOLVIMENTO	8
2.2.	EXTRACÇÃO DE PONTOS FORTES	10
2.2.1.	<i>Detector de Harris</i>	<i>10</i>
2.2.2.	<i>Métodos de extracção de pontos fortes considerados.....</i>	<i>12</i>
2.2.3.	<i>Resultados obtidos.....</i>	<i>13</i>
2.3.	EMPARELHAMENTO DE PONTOS FORTES ENTRE IMAGENS (<i>MATCHING</i>)	15
2.3.1.	<i>Correlação simples</i>	<i>16</i>
2.3.2.	<i>Correlação normalizada de média nula.....</i>	<i>17</i>
2.3.3.	<i>Algoritmo de Lucas-Kanade.....</i>	<i>17</i>
2.3.4.	<i>Métodos de emparelhamento de pontos considerados.....</i>	<i>18</i>
2.3.5.	<i>Resultados obtidos.....</i>	<i>19</i>
2.4.	GEOMETRIA EPIPOLAR.....	19
2.4.1.	<i>Método dos 8 pontos</i>	<i>24</i>
2.4.2.	<i>Método dos 7 pontos</i>	<i>24</i>
2.4.3.	<i>Algoritmo de RANSAC</i>	<i>24</i>
2.4.4.	<i>Algoritmo de LMedS.....</i>	<i>25</i>
2.4.5.	<i>Algoritmo de MAPSAC.....</i>	<i>26</i>
2.4.6.	<i>Determinação das linhas epipolares.....</i>	<i>26</i>
2.4.7.	<i>Métodos de cálculo da geometria epipolar considerados</i>	<i>26</i>
2.4.8.	<i>Resultados obtidos.....</i>	<i>27</i>
2.5.	RECTIFICAÇÃO	29
2.5.1.	<i>Rectificação projectiva sem cálculo da geometria epipolar</i>	<i>30</i>
2.5.2.	<i>Resultados obtidos.....</i>	<i>31</i>
2.6.	MAPA DE DISPARIDADE	32
2.6.1.	<i>Algoritmo de Stan Birchfield.....</i>	<i>34</i>
2.6.2.	<i>Resultados obtidos.....</i>	<i>35</i>
CAPÍTULO 3.	CONCLUSÕES E TRABALHO FUTURO	37
3.1.	CONCLUSÕES.....	38
3.2.	TRABALHO FUTURO.....	38
BIBLIOGRAFIA	39

Capítulo 1. Introdução

1.1. Introdução à Visão Computacional

Visão Computacional é a área da ciência que se dedica a desenvolver teorias e métodos e técnicas de extracção automática de informações úteis a partir de imagens (geométricas, topológicas, ...), de forma o mais semelhante possível à realizada pelo ser humano no seu complexo sistema cognitivo visual.

Tais imagens podem ser capturadas pelos mais diversos dispositivos, tais como câmaras de vídeo, *scanners*, etc. Por este motivo, a recente evolução tecnológica dos computadores digitais e dos dispositivos de aquisição de imagens, a preços cada vez mais reduzidos, tem possibilitado uma crescente aplicação da Visão Computacional nas mais diversas áreas [Coelho, 2003], como por exemplo:

- Engenharia (prototipagem rápida, *design*, ...);
- Realidade virtual (actores e objectos virtuais, sistemas de realidade aumentada, ...);
- Medicina (estudos antropométricos, análise da evolução de deformações, desenvolvimento e fabrico de próteses, planeamento de cirurgias, ...);
- Arquitectura (reconstrução de estruturas, esculturas, ...);
- Sistemas de navegação (controlo de *robots*, aquisição automática de alvos, ...);
- Indústria (controlo de qualidade, ...);
- Sistemas de segurança (identificação e autenticação de pessoas por reconhecimento da íris, impressões digitais, entre outras informações biométricas, ...).

É particularmente relevante ressaltar que, numa época em que a interacção com os computadores se torna cada vez mais frequente e necessária, a Visão Computacional permite considerar uma vasta área de interfaces mais eficazes, eficientes e intuitivas entre o mundo humano e o dos computadores, o que terá implicações revolucionárias tanto em termos tecnológicos e científicos como sociais.

Tendo como precursor Larry Roberts [Roberts, 1963], estudante de doutoramento no *MIT (Massachusetts Institute of Technology)* em 1962, foi só na década de 70 que se reconheceu a importância da Visão Computacional, principalmente no âmbito de determinação de parâmetros tridimensionais (3D). Com esta abertura, têm-se assistido, ao longo dos tempos ao aparecimento de novas técnicas que, directa ou indirectamente, permitiram obter das imagens mais informação, como o conhecimento das posições 3D

relativas a um determinado referencial ou orientações locais das superfícies captadas nas imagens [Shoaff, 2000].

As técnicas sem contacto para recuperar a geometria 3D de um objecto podem ser divididas em duas categorias [Coelho, 2003], Fig. 1:

1. *Activas* – requerem a projecção directa de energia (luz estruturada, ultra-sons, raios laser,...) sobre o objecto em análise ou utilizam o movimento da(s) câmara(s) ou do próprio objecto para obter informações sobre a sua forma 3D;
2. *Passivas* – não requerem a projecção de energia, usando apenas a iluminação ambiente existente, não necessitando de equipamentos especiais. São de aplicação mais geral mas, no entanto, a extracção de informação tridimensional é geralmente mais difícil.

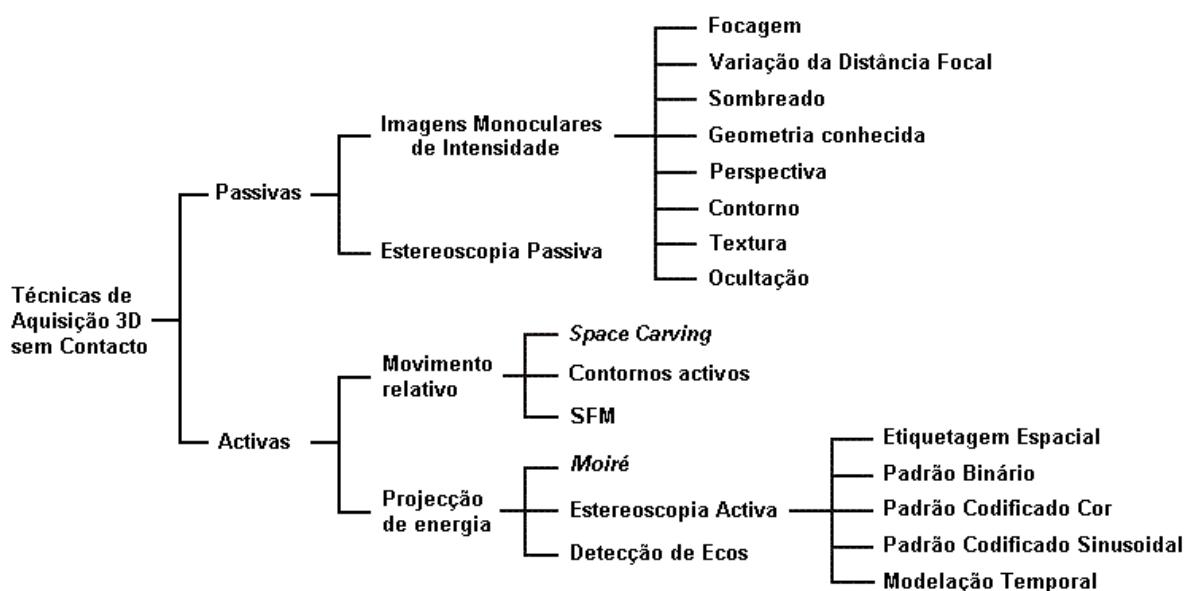


Fig. 1 - Divisão usual das técnicas sem contacto para obtenção de informação 3D de um objecto.

1.2. Reconstrução 3D do corpo humano

Nos dias de hoje, a reconstrução tridimensional do corpo humano é um tópico de grande interesse, com utilização prática na indústria cinematográfica, realidade virtual ou em aplicações biomédicas. Os modelos 3D são normalmente obtidos utilizando *scanners*, geralmente dispendiosos mas simples de usar. Tais sistemas, utilizam tecnologias (tais como: luz estruturada, feixe laser, ...) que providenciam um elevado número de pontos, utilizados

posteriormente para uma modelização do corpo humano de elevada qualidade e exactidão [Remondino, 2003]. Na Tabela 1 estão resumidas algumas das características de seis *scanners* comumente utilizados na reconstrução 3D do corpo humano.

Tabela 1 - Características de alguns *scanners* 3D (adaptado de [Remondino, 2003]).

	<i>Cyberware</i>	<i>TC2 Image Twin</i>	<i>Vitronics</i>	<i>Inspeck</i>	<i>Hamamatsu</i>	<i>Wicks&Wilson</i>
Produto	WB4, WBX	2T4s	Vitus	3D Full Body	64 BL Scanner	TriForm BS
Tempo de aquisição	~17s	~12s	<20s	-	<16s	~12s
Resolução	~1mm	~1mm	~1mm	~1.5mm	~1mm	~2mm
Tecnologia	laser	luz estruturada	laser	luz estruturada	luz estruturada	luz estruturada

Existem outras técnicas baseadas na extracção de silhuetas (como por exemplo, [Zheng, 1994]) ou em técnicas fotogramétricas (como por exemplo, [Remondino, 2003] e [D'Apuzzo, 2002]), que também têm sido alvo de considerável investigação nas últimas décadas.

1.3. Objectivos

O principal objectivo do trabalho inerente a este relatório, é a obtenção de modelos 3D de objectos utilizando a metodologia usualmente designada por *Reconstrução de Estruturas a partir do Movimento* (na literatura anglo-saxónica, *SFM - Structure from Motion*), que se insere no grupo das técnicas de Visão Activa [Pollefeys, 1998].

Proposta em 1979 por *Shimon Ullman* [Ullman, 1979], a técnica *SFM* recebeu, ao longo do tempo, contribuições e abordagens diversificadas. Dentro das possibilidades de variação, salienta-se o número de pontos de vista considerado sobre o objecto a reconstruir, a utilização ou não de pontos fortes e o conhecimento prévio ou não das características e posicionamento da(s) câmara(s).

No caso presente, não se pretende impor qualquer tipo de restrição ao movimento envolvido. Assim, partindo de uma sequência de imagens não calibradas de um objecto, pretende-se extrair o movimento relativo envolvido, a calibração da(s) câmara(s) considerada(s) e obter a geometria 3D do objecto em causa.

O método de *SFM* funciona essencialmente em dois passos distintos: a estimação do movimento no plano imagem e a obtenção da estrutura 3D do objecto e do movimento da(s) câmara(s).

Para estimar o movimento no plano imagem é necessário determinar onde é que os mesmos pontos 3D reais (pertencentes ao objecto a reconstruir) estão projectados bidimensionalmente nas várias imagens obtidas. Este é o problema da estimação da correspondência entre pontos de uma imagem e os pontos correspondentes nas imagens seguintes (emparelhamento).

Conhecendo as trajectórias dos pontos no plano da imagem, o segundo passo consiste em determinar a estrutura e o movimento 3D da(s) câmara(s), que melhor descreve a maioria das trajectórias estimadas.

Neste trabalho, começou-se por seguir a metodologia *SFM* proposta por *Faugeras* [Pollefeys, 2004], descrita na Fig. 2.

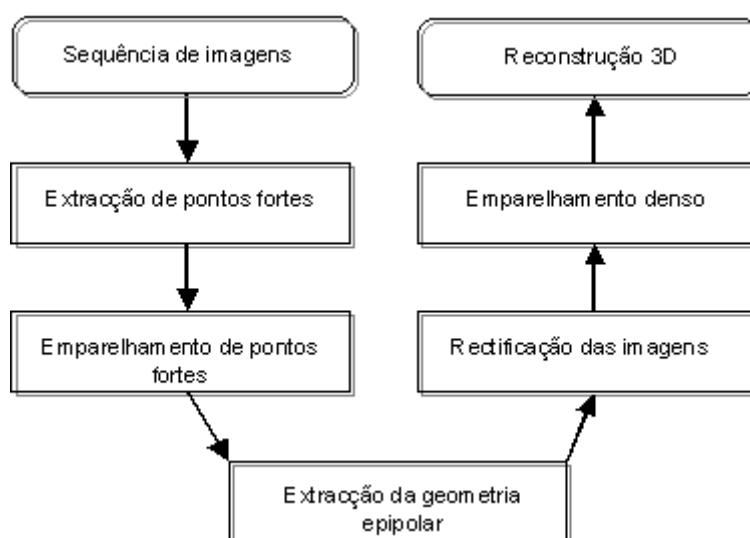


Fig. 2 – Estrutura da metodologia adoptada neste trabalho para a modelização 3D de objectos a partir de sequências de imagens.

Assim, de forma a conseguir tal objectivo, foi desenvolvida uma plataforma computacional modular na qual foram já integradas diversas funções disponibilizadas em várias bibliotecas de domínio público.

As funções integradas abrangem várias técnicas de Visão Activa, habitualmente usadas na obtenção da forma 3D de objectos a partir de sequências de imagens: extracção e emparelhamento de pontos fortes (pontos característicos) entre imagens, cálculo da geometria epipolar, rectificação e emparelhamento denso.

Neste relatório, após uma breve descrição da plataforma em desenvolvimento, são apresentados alguns resultados experimentais obtidos e, no final, são enumeradas algumas conclusões e indicadas perspectivas de trabalho futuro.

Capítulo 2. Plataforma Computacional

2.1. Plataforma Computacional em desenvolvimento

Ao longo deste trabalho, foi desenvolvida uma aplicação em C^{++} , com interface gráfica adequada para o utilizador e arquitectura modular, onde já se integraram algumas das funções disponibilizadas em cinco *softwares* e uma biblioteca computacional, todos de domínio público, para a reconstrução 3D de objectos usando técnicas de Visão Activa. Tais entidades serão neste relatório referidos genericamente como *Programas 1 a 6*:

- *Programa 1 – Peter’s Matlab Functions for Computer Vision and Image Analysis* [Kovesi, 2004]: pacote de funções em *Matlab* de processamento e análise de imagens e Visão 3D;
- *Programa 2 – Torr’s Matlab Toolkit* [Torr, 2002]: com uma interface gráfica em *Matlab*, aplica a técnica *SFM* entre duas imagens;
- *Programa 3 – OpenCV* [2004]: biblioteca de funções em C^{++} que implementam alguns dos algoritmos mais usuais no domínio da Visão Computacional;
- *Programa 4 – Kanade-Lucas-Tomasi (KLT) Feature Tracker* [Birchfield, 2004]: conjunto de funções em C , que implementam o algoritmo de *KLT*, para extrair e emparelhar pontos fortes numa sequência de imagens ([Lucas, 1981], [Tomasi, 1991]);
- *Programa 5 – Projective Rectification without Epipolar Geometry* [Isgrò, 1999]: programa em C , que realiza a rectificação de duas imagens estéreo sem predeterminar a geometria epipolar;
- *Programa 6 – Depth Discontinuities by Pixel-to-Pixel Stereo* [Birchfield, 1999]: programa em C , que retorna os mapas de disparidade e discontinuidades entre duas imagens rectificadas.

Os *programas* escritos originalmente em *Matlab* foram portados para C , utilizando a ferramenta *MATLAB Compiler*, de forma a serem convenientemente integrados na plataforma computacional em desenvolvimento.

A referida plataforma foi desenvolvida numa forma modular, permitindo ao utilizador facilmente escolher, em cada uma das técnicas de Visão Activa disponibilizadas, o algoritmo que pretende usar, Fig. 3, assim como definir comodamente os seus parâmetros, Fig. 4. Esta característica permite facilmente controlar os parâmetros de cada algoritmo utilizado, analisar

e comparar os resultados obtidos por diferentes técnicas, assim como possibilita a expansão da plataforma com a fácil integração de novas técnicas e/ou algoritmos.

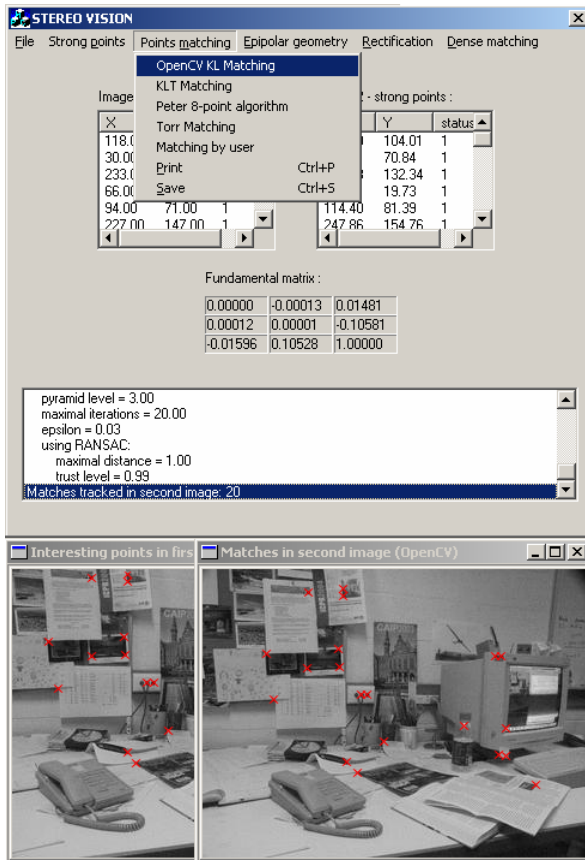


Fig. 3 - Em cada item do menu da plataforma, são apresentados os vários algoritmos disponíveis para cada técnica; em baixo, duas imagens resultantes da aplicação do algoritmo de emparelhamento de pontos fortes do *programa OpenCV*.

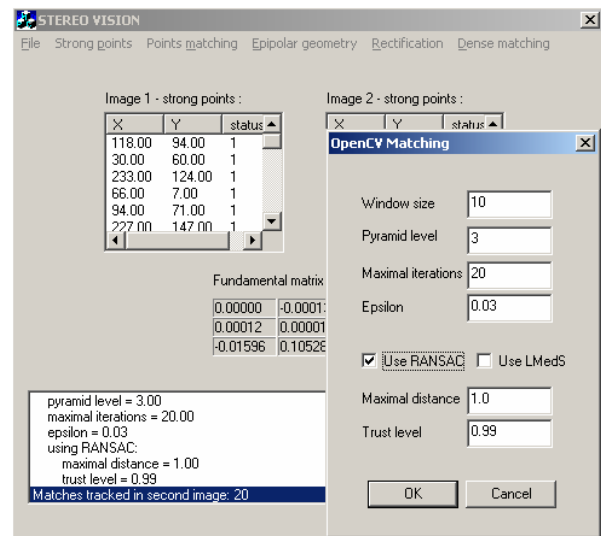


Fig. 4 - Exemplo de selecção dos parâmetros do algoritmo de emparelhamento de pontos fortes do *programa OpenCV*, disponível na plataforma.

A modularidade da referida plataforma foi conseguida, uniformizando determinados parâmetros das funções integradas, o que implicou, por vezes, adaptar o código original dos *programas* computacionais considerados.

Para os testes experimentais realizados, utilizaram-se imagens de dimensões 540×612 píxeis, obtidas por uma câmara fotográfica digital convencional.

De seguida, vão ser apresentadas as técnicas de Visão 3D consideradas e analisadas, incluindo uma breve descrição da sua implementação, nos vários *programas* considerados, assim como alguns dos resultados obtidos pelas mesmas.

2.2. Extracção de pontos fortes

O primeiro passo dos métodos de Visão 3D mais comuns é a detecção de pontos fortes numa imagem. Esses pontos são aqueles que têm uma componente 2D característica sobre o objecto; ou seja, são pontos que reflectem discrepâncias relevantes entre os seus valores de intensidades e os dos seus vizinhos; usualmente, são os vértices dos objectos. A sua extracção, permite posteriormente correlacioná-los sequencialmente noutras imagens e este emparelhamento torna possível a posterior reconstrução 3D.

2.2.1. Detector de Harris

Numa imagem I em tons de cinzento, a matriz de covariância do gradiente, M , define-se pela seguinte equação:

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}, \quad (1)$$

sendo $A = (\partial I / \partial x)^2$, $B = (\partial I / \partial y)^2$ e $C = (\partial I / \partial x)(\partial I / \partial y)$, com $(\partial I / \partial x)$ e $(\partial I / \partial y)$ os gradientes da imagem I nas direcções x e y , respectivamente, Fig. 5.

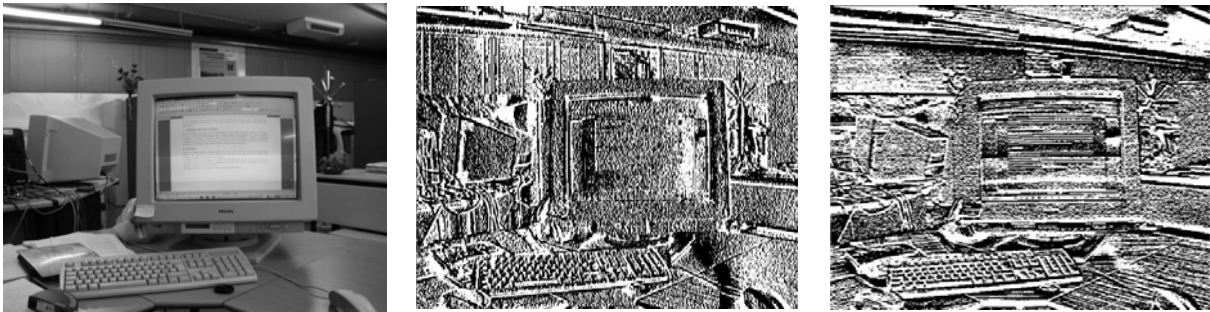


Fig. 5 – Da esquerda para a direita: imagem original I , resultado da derivada na direcção horizontal $(\partial I / \partial x)$ e vertical $(\partial I / \partial y)$.

Se os dois valores próprios da matriz M , λ_1 e λ_2 , forem elevados e similares em magnitude, significa que foi detectado um vértice, Fig. 6.

Em 1988, *Harris e Stephens* [Harris, 1988], definiram a seguinte função de medida para a determinação de pontos fortes:

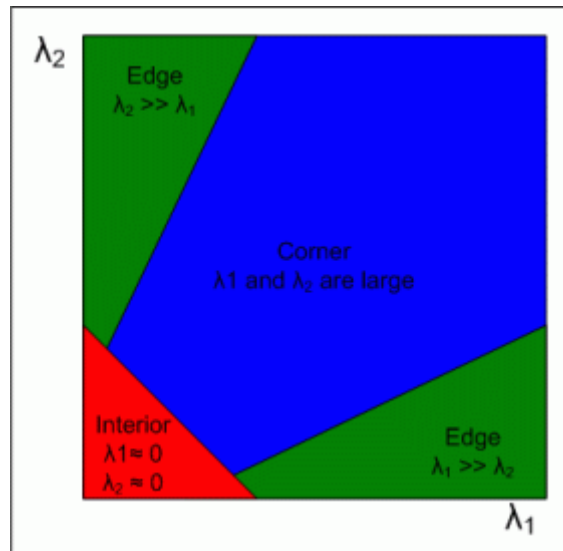


Fig. 6 - Divisão do espaço dos valores próprios de M , para a detecção de pontos fortes ou fronteiras de objectos (imagem retirada de [Parks]).

$$R = \det(M) - k(\text{trace}(M))^2, \quad (2)$$

onde $\det(M) = \lambda_1 \lambda_2 = AB - C^2$ e $\text{trace}(M) = \lambda_1 + \lambda_2 = A + B$. Harris sugere $k = 0.04$, mas como usualmente este parâmetro necessita de ser estimado, Noble [Noble, 1989] sugeriu um re-arranjo da função de medida:

$$R = \det(M) / (\text{trace}(M) + \varepsilon). \quad (3)$$

em que ε é uma constante (2×10^{-52}) utilizada de forma a evitar um denominador nulo no caso de M ter traço nulo.

Na Fig. 7, pode-se verificar o resultado da aplicação das funções de medida (2) e (3), sobre uma imagem em tons de cinzento.

Em ambos os casos, se os valores próprios forem similares e elevados, $\det(M)$ é muito maior que $\text{trace}(M)$, logo o resultado R será também elevado. Os máximos locais de R serão os pontos fortes da imagem I . De forma a minimizar os resultados obtidos, pode-se impor que R seja superior a um dado valor de limiar (*threshold*).

Como este método é muito sensível ao ruído, usualmente aplica-se aos termos de M um filtro de suavização Gaussiano. Como este filtro introduz pontos fortes extra, tal pode ser compensado utilizando posteriormente um operador morfológico de dilatação.

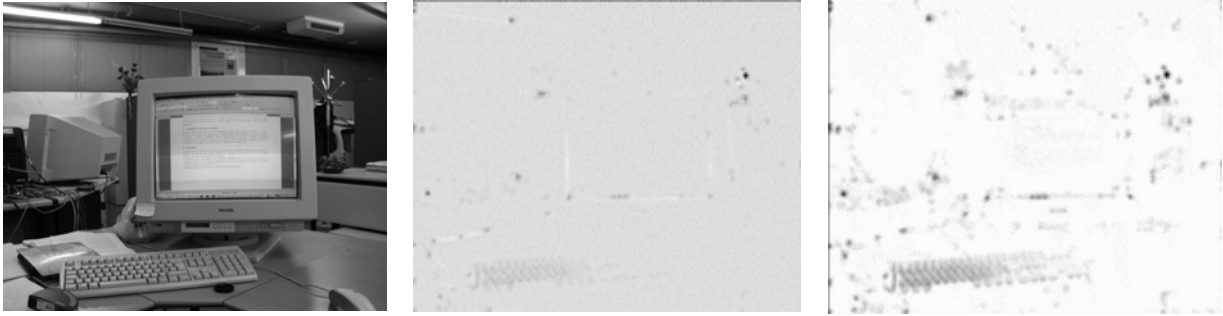


Fig. 7 – Da esquerda para a direita: imagem original, resultado da aplicação da equação (2) (imagem negada) e da equação (3) (imagem negada).

2.2.2. Métodos de extracção de pontos fortes considerados

Na Fig. 8, estão visíveis os algoritmos actualmente disponíveis na plataforma computacional para a detecção de pontos fortes.

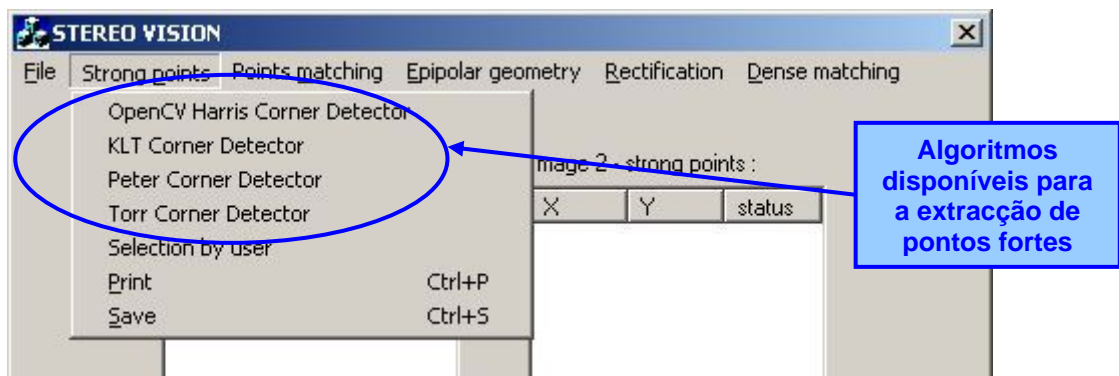


Fig. 8 – Algoritmos disponíveis, na plataforma computacional, para a determinação de pontos fortes numa imagem.

Os programas 1 e 2 utilizam o método de Harris. Ambos calculam M , utilizando o operador de gradiente Prewitt, G :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}. \quad (4)$$

O programa 1, utiliza a equação (3) para definir a função de medida R e determina como pontos fortes (x, y) aqueles que satisfaçam a condição de *threshold* $R(x, y) \geq t$ e a condição $R(x, y) = D(x, y)$, sendo D a imagem após a dilatação de R .

Já o programa 2, usa a equação (2) com $k = 0.04$, elimina todos os píxeis que não são máximos numa janela de dimensão 3×3 , e selecciona os n pontos mais fortes.

Por outro lado, os *programas 3 e 4* determinam os pontos fortes directamente dos valores próprios de M , utilizando para tal o operador de gradiente de *Sobel*:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (5)$$

Para píxel (x, y) da imagem I , determina $\lambda = \min(\lambda_1, \lambda_2)$, seleccionando os máximos numa vizinhança $w \times w$ e eliminando os que não satisfaçam a condição de *threshold* $\lambda \geq t$.

O *programa 3* selecciona os pontos (x, y) que satisfaçam a condição $\lambda(x, y) = D(x, y)$, sendo $D(x, y)$ a dilatação de $\lambda(x, y)$, e no final garante uma distância mínima d entre os pontos fortes determinados.

Por seu lado, o *programa 4* também garante uma distância mínima d entre os pontos fortes encontrados e, no final, selecciona os n mais fortes.

2.2.3. Resultados obtidos

Nas Fig. 9 a Fig. 11, estão apresentados os resultados obtidos pelos *programas 1 a 4*, na extracção de pontos fortes nas imagens de teste consideradas.

Como se pode observar, o *programa 2 (Torr's Matlab Toolkit)* é o que apresenta piores resultados, principalmente pelo aparecimento de zonas com uma densidade muito elevada de pontos característicos, pois não garante uma distância mínima entre estes, tal como acontece com os *programas 3 (OpenCV)* e *4 (KLT)*. Esta propriedade pode ser relevante numa escolha criteriosa de pontos em zonas da imagem com variações de gradiente mais acentuadas.

Embora o *programa 1 (Peter's Matlab Functions)* não imponha uma distância mínima entre pontos, tal efeito é conseguido com o uso do operador morfológico de dilatação, obtendo-se assim resultados bastante razoáveis.

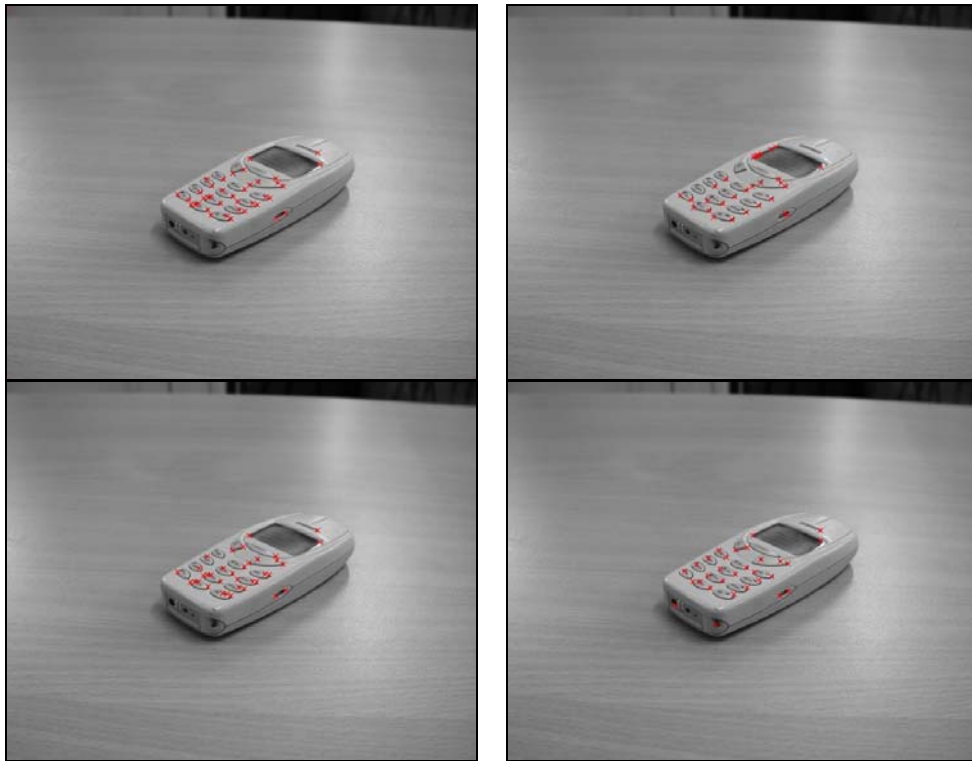


Fig. 9 - As cruzes vermelhas assinalam os pontos fortes encontrados na imagem *Tlm_1* (da direita para esquerda e de cima para baixo, são os resultados dos *programas 1 a 4*).

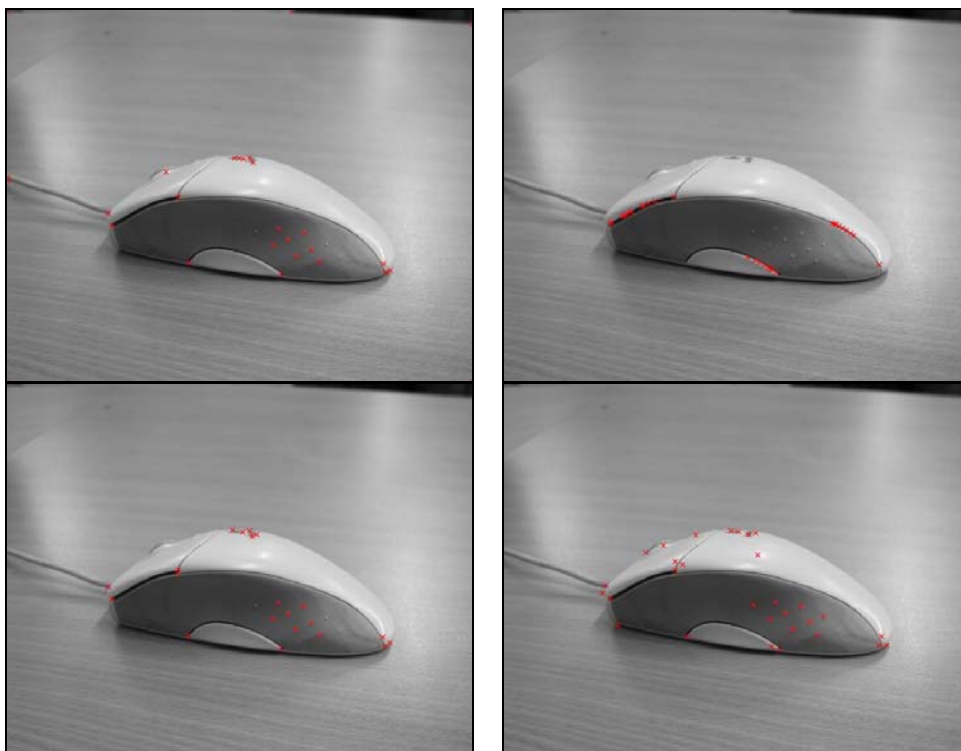


Fig. 10 - As cruzes vermelhas assinalam os pontos fortes encontrados na imagem *Rato_1* (da direita para esquerda e de cima para baixo, são os resultados dos *programas 1 a 4*).

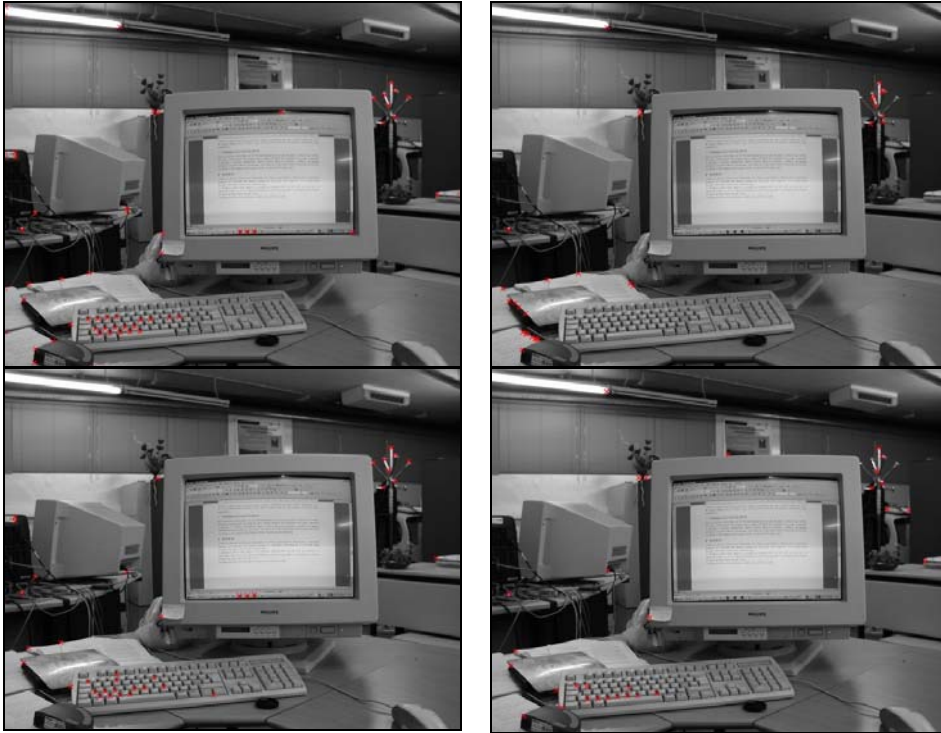


Fig. 11 - As cruzes vermelhas assinalam os pontos fortes encontrados na imagem *Pc_1* (da direita para esquerda e de cima para baixo, são os resultados dos *programas 1 a 4*).

Após a determinação dos pontos fortes, a plataforma apresenta ao utilizador a lista das coordenadas dos pontos fortes encontrados na primeira imagem do par estéreo, o algoritmo utilizado e os respectivos parâmetros de execução, Fig. 12.

2.3. Emparelhamento de pontos fortes entre imagens (*matching*)

Na metodologia de reconstrução 3D considerada é necessário identificar os pontos 2D nas várias imagens que resultem da projecção do mesmo ponto da cena (*matching*). Geralmente um reduzido número de pontos de correspondência, é suficiente para se poder determinar uma relação geométrica entre duas imagens, ou seja, a matriz fundamental.

A detecção automática das correspondências existentes entre os pontos fortes de duas imagens parcialmente sobrepostas, pode ser realizada através do cálculo de uma das várias medidas de *matching* existentes, medidas essas que, em geral, para cada ponto forte numa imagem *I*, calculam o grau de semelhança em toda uma imagem *J* ou para todos os pontos fortes da imagem *J*, sendo para isso analisada e comparada a envolvência de cada um desses pontos.

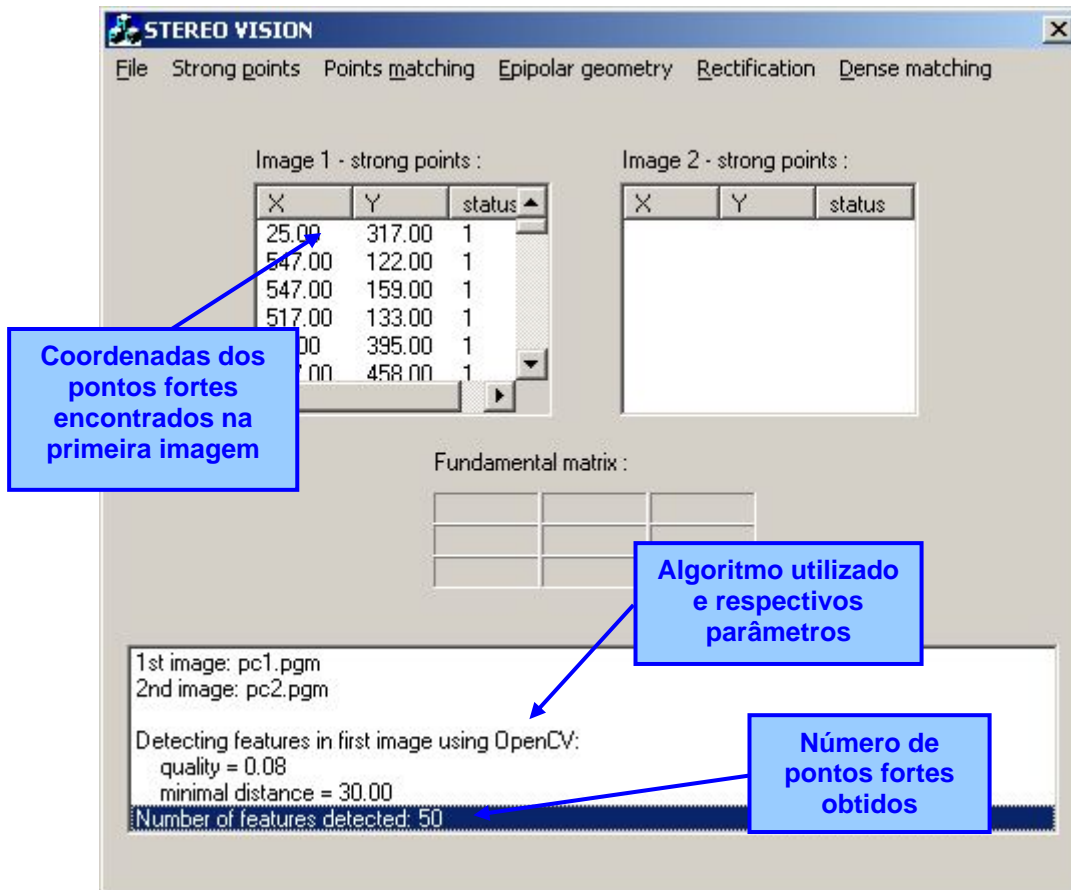


Fig. 12 – Interface da plataforma computacional após a detecção de pontos fortes na primeira imagem.

2.3.1. Correlação simples

Este método é usualmente conhecido como *SSD* (*sum-of-square-differences*). Considere-se o ponto $m = [m_x \ m_y]$ da imagem I . O objectivo é encontrar o ponto $m' = m + d = [m_x + d_x \ m_y + d_y]$ na imagem J tal que $I(m)$ e $J(m')$ sejam pontos similares. Tal obtém-se minimizando a função de dissimilaridade, definida como:

$$D(d) = \iint_w (I(x, y) - J(x + d_x, y + d_y))^2. \quad (6)$$

Um *matching* perfeito corresponde a obter $D(d) = 0$. Note-se que esta medida é efectuada numa região de tamanho $w \times w$, denominada por janela de integração.

Dado se tratar de uma medida não normalizada, este método é sensível a diferenças de iluminação das duas imagens I e J , assim como é significativamente afectada por diferenças de contraste nas regiões de interesse.

2.3.2. Correlação normalizada de média nula

Os problemas encontrados no método anterior são superados se se normalizarem os vectores referentes às imagens, resultando no seguinte coeficiente de correlação:

$$S(d) = \frac{\iint_w (I(x, y) - \bar{I}) \cdot (J(x + d_x, y + d_y) - \bar{J})}{\sqrt{\iint_w (I(x, y) - \bar{I})^2} \cdot \sqrt{\iint_w (J(x + d_x, y + d_y) - \bar{J})^2}}, \quad (7)$$

onde \bar{I} e \bar{J} são os valores médios nas regiões de interesse definidas por w . Este método é denominado por *ZNCC* (*zero-mean normalized cross-correlation*). Um *matching* perfeito corresponde a obter $S(d)=1$ e quanto maior for este valor, mais forte é a correspondência entre as regiões de interesse nas duas imagens.

2.3.3. Algoritmo de Lucas-Kanade

Este algoritmo assume que, se o vector de deslocamento $d = [d_x \quad d_y]$ for reduzido, a minimização da equação (6) resume-se a resolver o seguinte sistema de duas equações a duas incógnitas:

$$d = e \cdot M^{-1}, \text{ com } e = \begin{bmatrix} \sum_w I_x I_t \\ \sum_w I_y I_t \end{bmatrix}, \quad (8)$$

sendo M a matriz de covariância do gradiente da primeira imagem (equação (1)), $I_x = \partial I / \partial x$, $I_y = \partial I / \partial y$ e $I_t = I(x, y) - J(x, y)$.

Este algoritmo, é utilizado de uma forma iterativa: começa por obter uma estimativa de d a partir de imagens suavizadas de baixa resolução (imagens piramidais), refinando os pontos de emparelhamento com imagens de resolução cada vez mais elevada, até que na última iteração se utilizam as imagens originais ([Lucas, 1981], [Bouguet, 1999]), Fig. 13.



Fig. 13 - Exemplo de construção de uma estrutura piramidal com 3 níveis, a partir da imagem original (à esquerda).

2.3.4. Métodos de emparelhamento de pontos considerados

Após a determinação dos pontos fortes na primeira imagem, as funções de emparelhamento tornam-se disponíveis para o utilizador, Fig. 14.

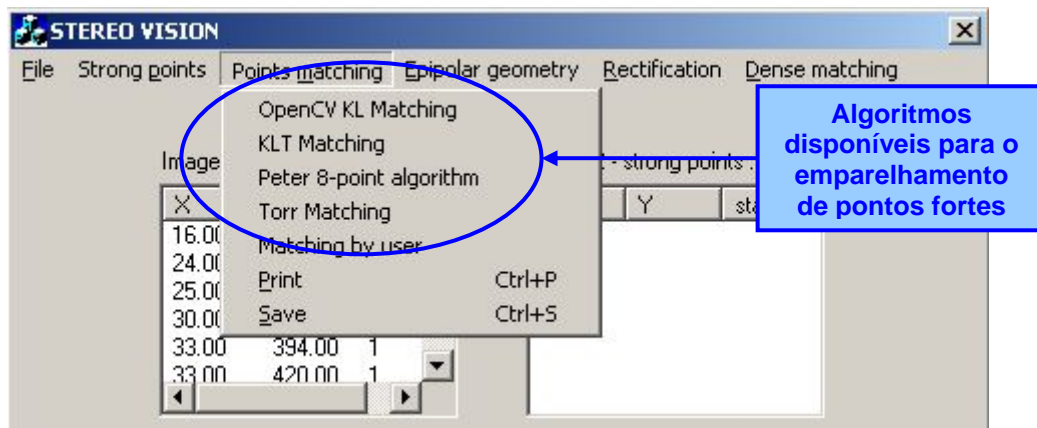


Fig. 14 - Algoritmos disponíveis, na plataforma computacional, para o emparelhamento de pontos fortes entre um par de imagens estéreo.

O *programa 1* utiliza a medida de similaridade do método *ZNCC*, dada pela equação (7), e o *programa 2*, a medida de dissimilaridade do método *SSD*, dada pela equação (6). Ambos têm como parâmetros de entrada os pares de pontos fortes (m, m') , obtidos nas duas imagens estéreo I e J , aplicando a função de medida para cada par que satisfaça a condição $\|m - m'\| \leq d^2$; ou seja, determinam um desvio máximo entre pontos de correspondência nas duas imagens, tornando assim menor o esforço computacional (o valor d deve ser cuidadosamente escolhido).

Já os *programas 3* e *4* utilizam o método de *Lucas-Kanade* (ver secção 2.3.3), necessitando somente do conjunto de pontos fortes m encontrados na primeira imagem, do número de imagens piramidais n a usar, do número de iterações máximas para tentar

encontrar o vector de deslocamento d e do valor limite permitido para o resíduo e (equação (8)).

2.3.5. Resultados obtidos

De forma a garantir uma comparabilidade mais adequada dos resultados obtidos pelas técnicas consideradas na plataforma para o emparelhamento de pontos fortes, foi utilizado o *programa 4 (KLT)* para seleccionar os 30 pontos fortes na primeira imagem, que são utilizados como parâmetros de entrada nesta fase da metodologia.

Os resultados experimentais obtidos são apresentados nas Fig. 15 a Fig. 17. Nestes, pode-se observar que o emparelhamento de pontos fortes é uma fase crítica em Visão 3D.

Em todos algoritmos considerados, verificam-se erros na correlação, sendo o *programa 2 (Torr's Matlab Toolkit)* o que apresenta os piores resultados.

Em relação às imagens de teste utilizadas, o pior caso é com as imagens “Rato”, pois o objecto em causa apresenta contornos curvilíneos e sem fortes contrastes, factores que complicam em muito o processo de emparelhamento.

Após o emparelhamento dos pontos fortes, a aplicação apresenta ao utilizador a lista das coordenadas dos pontos obtidos na segunda imagem, Fig. 18.

2.4. Geometria epipolar

Dado um par de imagens estéreo, qualquer ponto P do espaço 3D define um plano π_p que contém o ponto P e os centros ópticos O_l e O_r de ambas as câmaras. O plano π_p é definido como *plano epipolar* e as linhas de intersecção deste plano com os planos das imagens são definidas como *linhas epipolares conjugadas* [Hartley, 2004], Fig. 19.

A geometria epipolar corresponde à estrutura geométrica entre duas vistas e expressa-se matematicamente pela matriz fundamental F , de dimensão 3×3 :

$$m'^T F m = 0, \quad (9)$$

onde m e m' são os pontos de correspondência entre duas imagens. O ponto m , está sobre a linha epipolar l , definida por:

$$l = F m'. \quad (10)$$

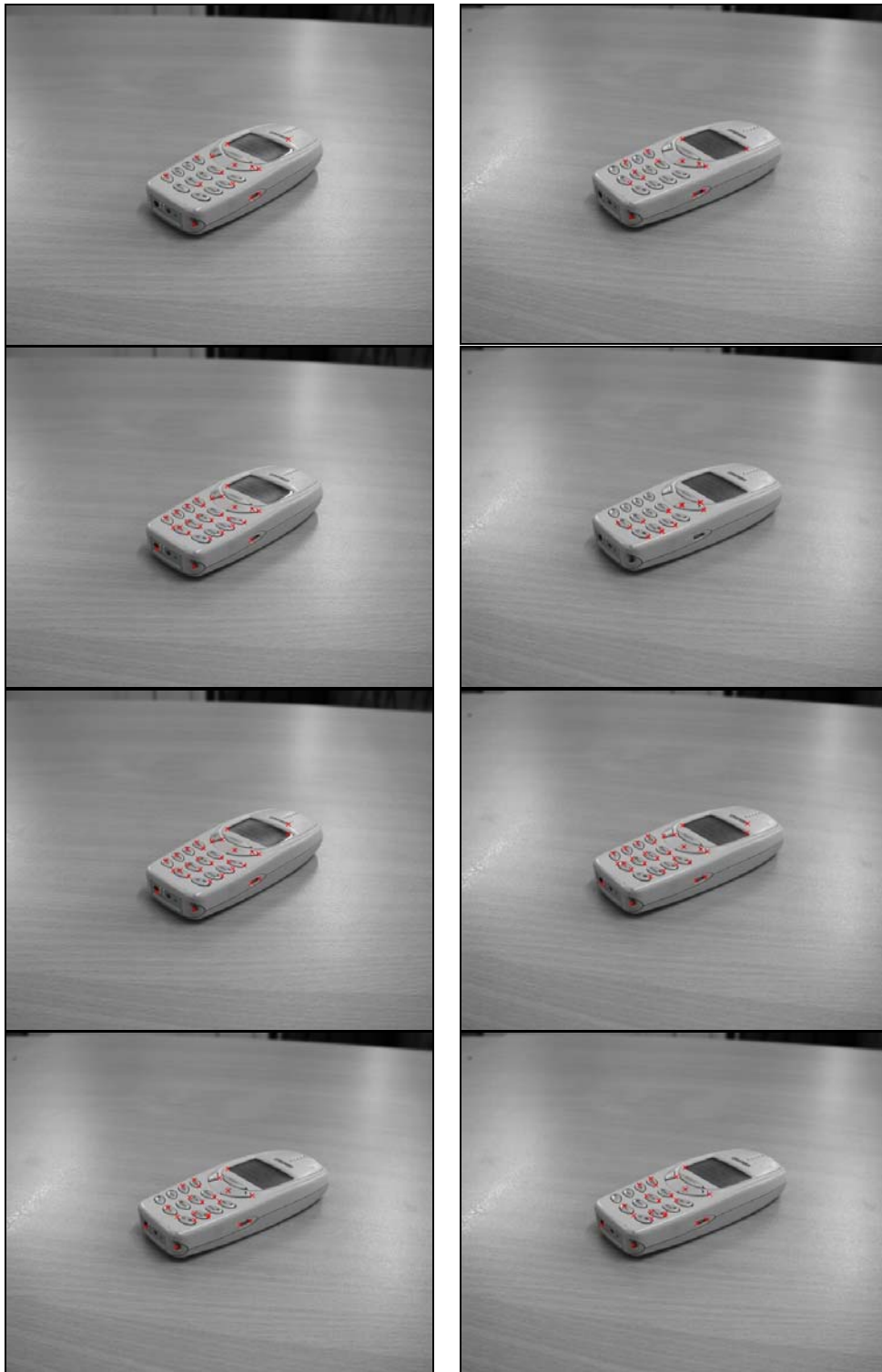


Fig. 15 - Na coluna da direita a imagem *Tlm_1*, e na da esquerda a imagem *Tlm_2*. As cruzes vermelhas assinalam os pontos emparelhados (de cima para baixo, por linha, são os resultados dos *programas 1 a 4*).

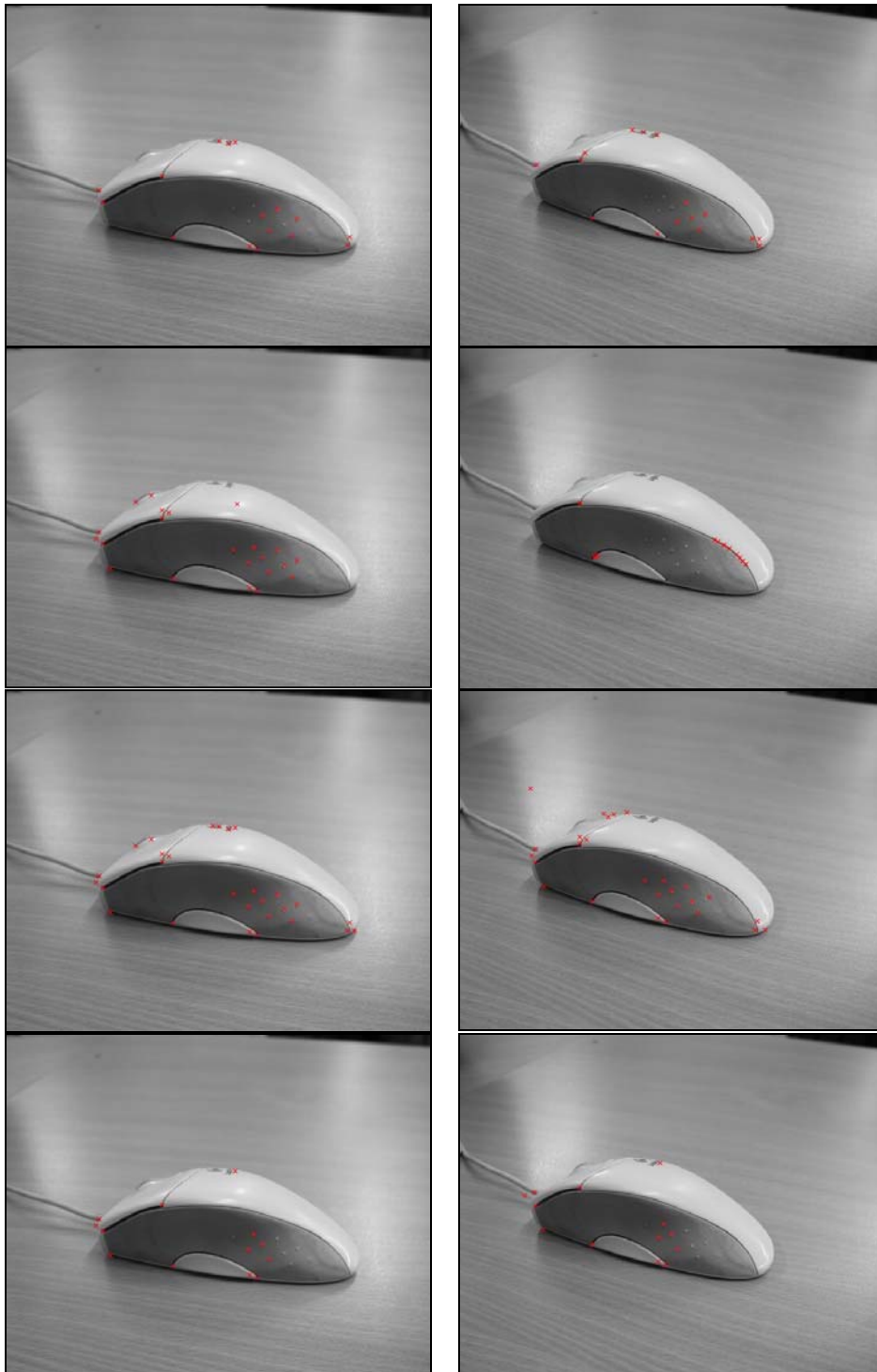


Fig. 16 - Na coluna da direita a imagem *Rato_1* e na da esquerda a imagem *Rato_2*. As cruces vermelhas assinalam os pontos emparelhados (de cima para baixo, por linha, são os resultados dos *programas 1 a 4*).



Fig. 17 - Na coluna da direita a direita a imagem *Pc_1* e na da esquerda a imagem *Pc_2*. As cruzes vermelhas assinalam os pontos emparelhados (de cima para baixo, por linha, são os resultados dos *programas 1 a 4*).

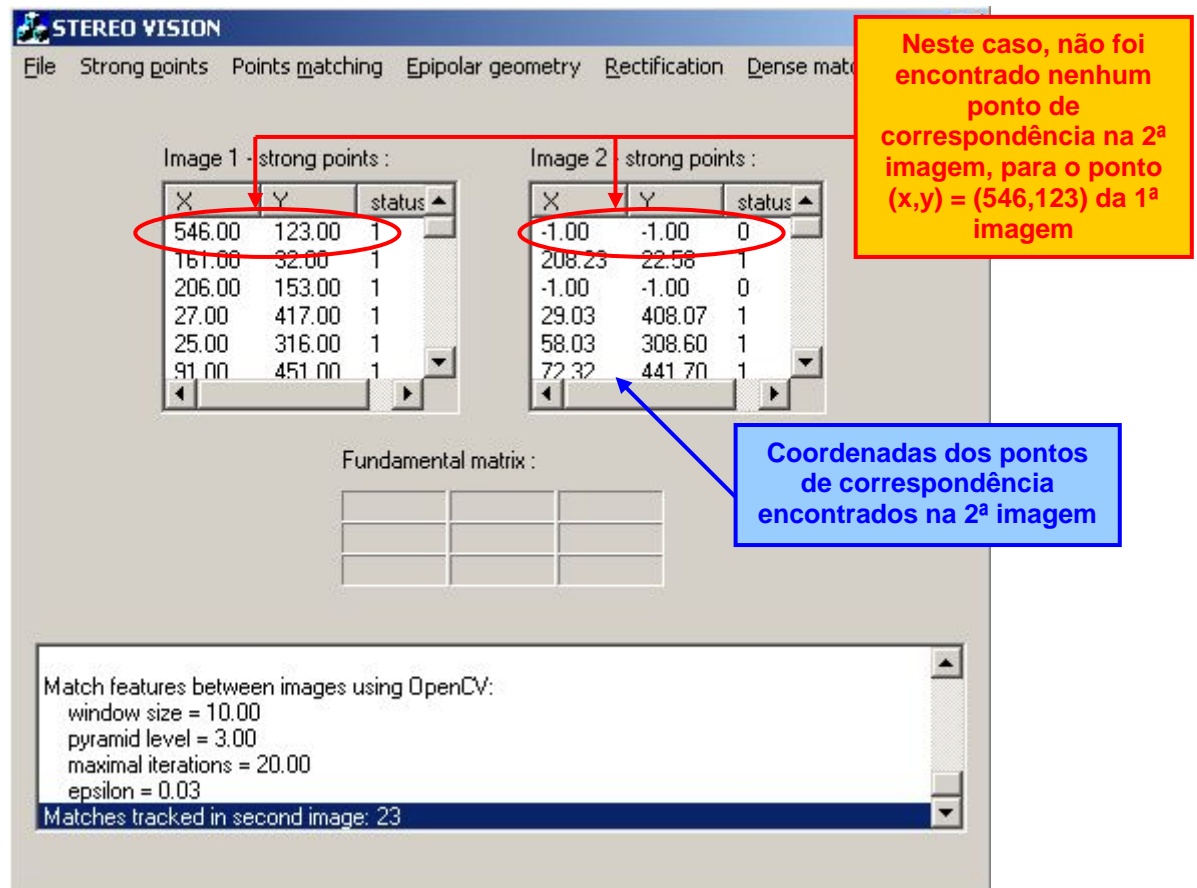


Fig. 18 - Interface da plataforma computacional após o emparelhamento de pontos fortes entre duas imagens.

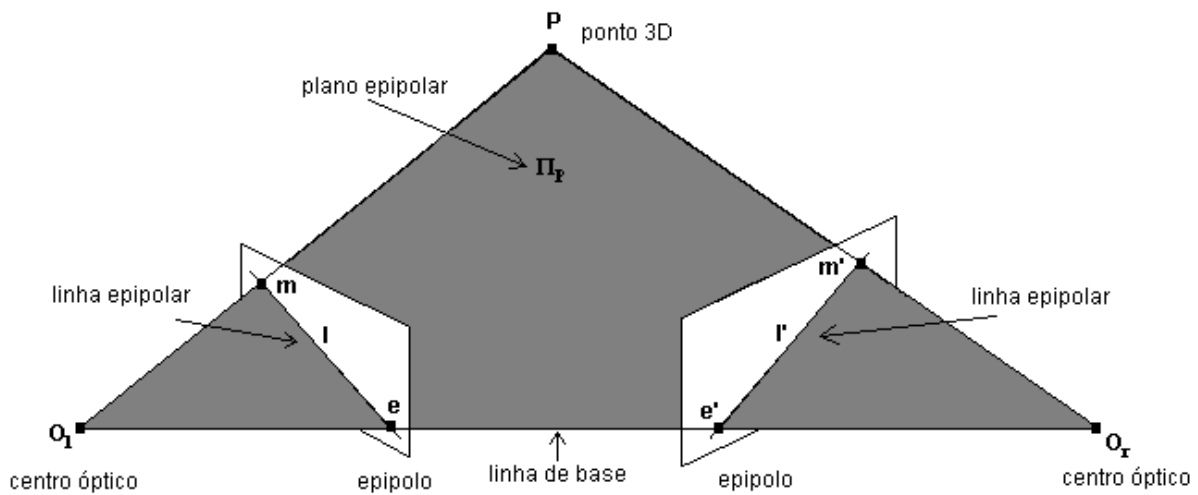


Fig. 19 – Geometria epipolar.

Assim, o cálculo da geometria epipolar significa obter informação da pose relativa entre duas vistas da mesma cena. Tal informação permite também eliminar erros no emparelhamento efectuado previamente, assim como facilitar a obtenção de novos pontos de correlação (emparelhamento denso).

2.4.1. Método dos 8 pontos

Como $m = [x \ y \ 1]^T$ e $m' = [x' \ y' \ 1]^T$, pode-se rearranjar a equação (9), obtendo-se:

$$[xx' \ yx' \ x' \ xy' \ yy' \ y' \ x \ y \ 1]F = 0. \quad (11)$$

Com esta equação matricial, F pode ser determinada de forma linear pelo método *SVD – Singular Value Decomposition* [Golub, 1983] utilizando 8 ou mais pares de pontos emparelhados.

Como o sistema de equações (11) é homogêneo, a solução obtida é única a menos de um factor de escala [Hartley, 1997].

2.4.2. Método dos 7 pontos

Utilizando somente 7 pares de pontos emparelhados, é possível resolver a equação (11) impondo a restrição de a característica de F ser igual a 2 (o sistema de equações torna-se não linear).

Tal como para o algoritmo de 8 pontos, é aconselhado normalizar as coordenadas da imagem antes da determinação de F [Hartley, 1997].

2.4.3. Algoritmo de RANSAC

RANSAC – RANdom SAMpling Consensus foi proposto em 1981 por *Fischler* e *Bolles* [Fischler, 1981] e possibilita o cálculo robusto de F , permitindo distinguir pontos de emparelhamento verdadeiros e falsos (*inliers* e *outliers*, respectivamente). Assim, partindo de um subconjunto de pontos m e m' , calcula-se uma primeira estimativa de F utilizando um dos métodos anteriores, e com esta determinam-se os *inliers* e *outliers* do restante conjunto de pontos. Quanto menor o subconjunto inicial, menor é a probabilidade de este estar “contaminado” por *outliers*. Este processo, repete-se até se ter atingido um certo grau de confiança: por exemplo, a probabilidade Γ (usualmente estabelecida como igual a 95%) de se ter escolhido um subconjunto de *inliers* após n iterações:

$$\Gamma = 1 - (1 - (1 - \varepsilon)^p)^n, \quad (12)$$

sendo ε a fracção dos *outliers* e p o número de pontos contidos no subconjunto [Rousseeuw, 1987]. Os pontos são considerados *outliers* se estiverem demasiado afastados das linhas epipolares.

A Tabela 2 esquematiza sumariamente o algoritmo de *RANSAC*.

Tabela 2 - Resumo do algoritmo de *RANSAC*.

<p><i>Passo 1:</i> Enquanto $\Gamma(\#inliers, \#outliers) < 95\%$:</p> <p style="padding-left: 40px;"><i>Passo 1.1:</i> seleccionar aleatoriamente 7 ou 8 pontos de <i>matching</i>;</p> <p style="padding-left: 40px;"><i>Passo 1.2:</i> calcular a matriz fundamental F;</p> <p style="padding-left: 40px;"><i>Passo 1.3:</i> determinar os pontos <i>inliers</i>;</p> <p><i>Passo 2:</i> Refinar a matriz F considerando todos os pontos <i>inliers</i>.</p>
--

2.4.4. Algoritmo de *LMedS*

O algoritmo *LMedS* – *Least-Median-of-Squares* ([Rousseeuw, 1984]) difere do anterior no facto de a melhor estimação ser aquela que minimiza a média dos erros r_i , definidos por:

$$r_i^2 = d^2(m_i', Fm_i) + d^2(m_i, F^T m_i'). \quad (13)$$

Neste algoritmo, a cada correspondência é atribuído o peso w_i :

$$w_i = \begin{cases} 1 & r_i^2 \leq t^2 \\ 0 & otherwise \end{cases}. \quad (14)$$

A matriz F é então calculada resolvendo o seguinte problema de minimização:

$$\min \sum_i w_i r_i^2. \quad (15)$$

2.4.5. Algoritmo de MAPSAC

MAPSAC - *Maximum A Posteriori Sample Consensus* ([Torr, 2002]) redefine a equação (14) de forma a atribuir uma “penalidade” r_i^2 aos *outliers* e um “peso” t^2 aos *inliers*, consoante estes se adequam melhor ou pior à geometria do sistema:

$$w_i = \begin{cases} r_i^2 & r_i^2 \leq t^2 \\ t^2 & \text{otherwise} \end{cases} \quad (16).$$

2.4.6. Determinação das linhas epipolares

Após ter sido determinada a matriz fundamental F , a partir do emparelhamento de pontos fortes, a linha epipolar l que passa pelo ponto m na imagem I pode ser obtida pela equação (10).

2.4.7. Métodos de cálculo da geometria epipolar considerados

Na Fig. 20, estão visíveis os algoritmos considerados para o cálculo da matriz fundamental.

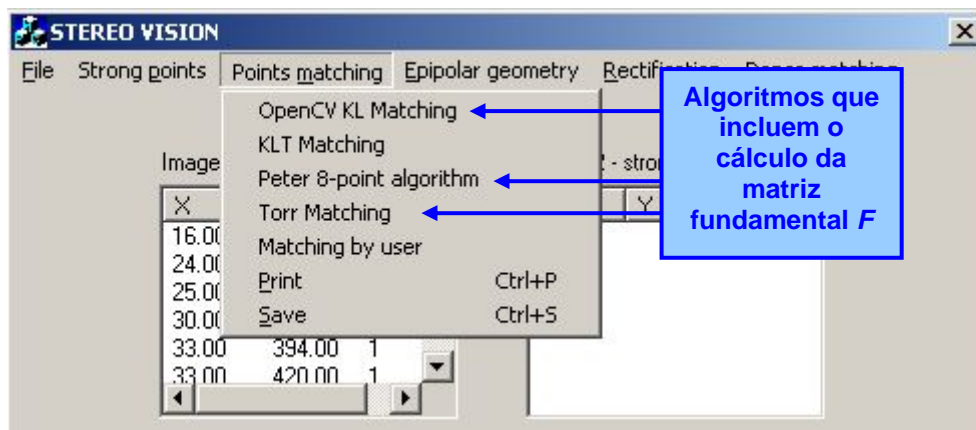


Fig. 20 - Algoritmos disponíveis, na plataforma computacional, para o cálculo da matriz fundamental F .

O programa *I* calcula a geometria epipolar usando o algoritmo de RANSAC, utilizando o método de 8 pontos para o cálculo das estimativas de F em cada iteração. Termina quando $\Gamma \geq 99\%$ (12), e recalcula F , novamente pelo método de 8 pontos, utilizando somente os pontos *inliers* encontrados.

Já o *programa 3*, procede de igual forma mas fornece a possibilidade de optar entre o algoritmo de *RANSAC* ou *LMedS*.

Por seu lado, o *programa 2* utiliza o algoritmo de *MAPSAC*, sendo as estimativas iniciais de F calculadas pelo método de 7 pontos. No final, após a determinação dos pontos *inliers* e *outliers*, F é determinada pelo método de 8 pontos, tendo o cuidado de minimizar o erro introduzido pela presença de *outliers* [Torr, 2002].

Somente os *programas 1* e *3* permitem representar, nas imagens dos resultados, as linhas epipolares obtidas pelos mesmos.

2.4.8. Resultados obtidos

O cálculo da matriz fundamental F (como uma opção extra nos parâmetros de cada função, Fig. 21) encontra-se incluído no menu de emparelhamento, pois, como foi descrito anteriormente, os *programas* determinam os pontos *inliers* e *outliers* ao mesmo tempo que calculam F .

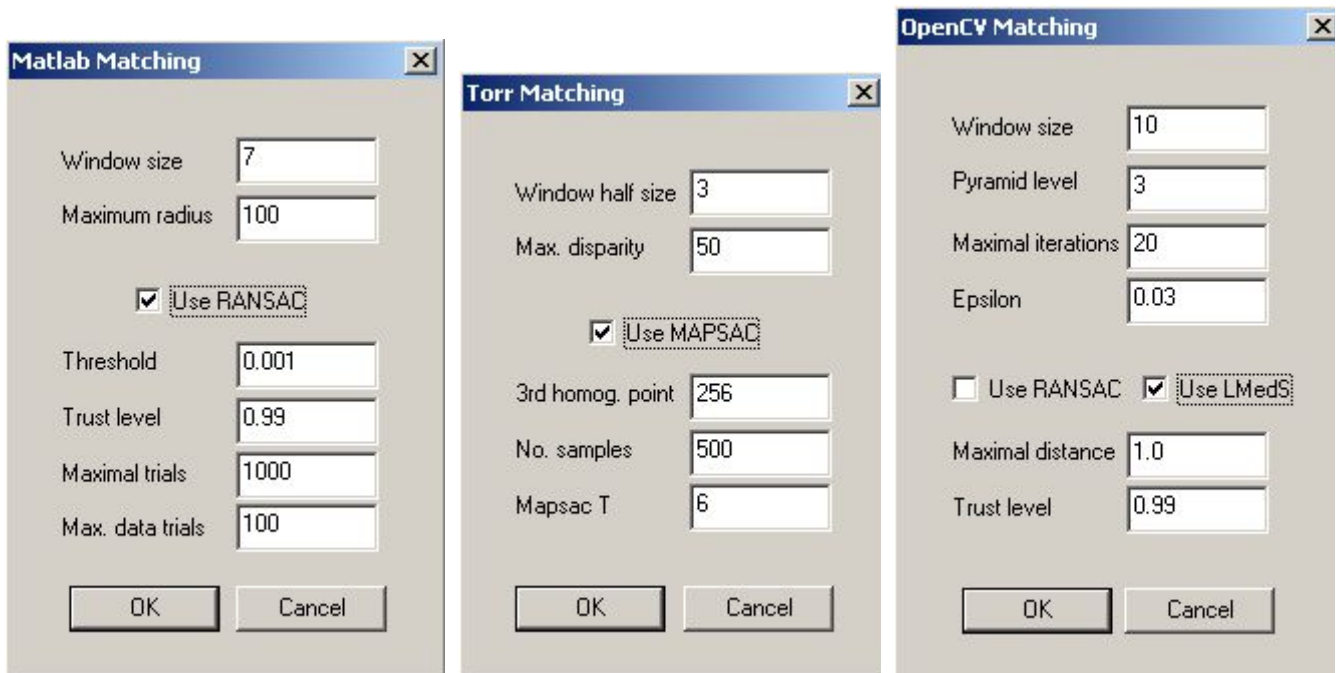


Fig. 21 - Opção extra de determinação da matriz fundamental F , inserida nas funções de emparelhamento de pontos entre imagens (da direita para a esquerda: *programas 1, 2 e 3*).

Após o cálculo da matriz fundamental F , com a determinação dos pontos *inliers* e *outliers*, torna-se possível ao utilizador proceder à determinação das linhas epipolares nos pontos *inliers*, Fig. 22.

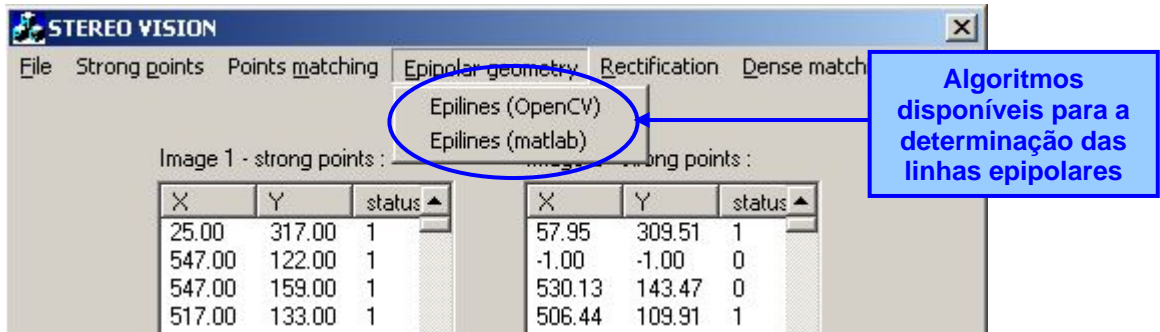


Fig. 22 - Algoritmos disponíveis, na plataforma computacional, para a determinação das linhas epipolares.

A Fig. 23, apresenta os resultados obtidos pelos algoritmos considerados para a determinação das linhas epipolares.

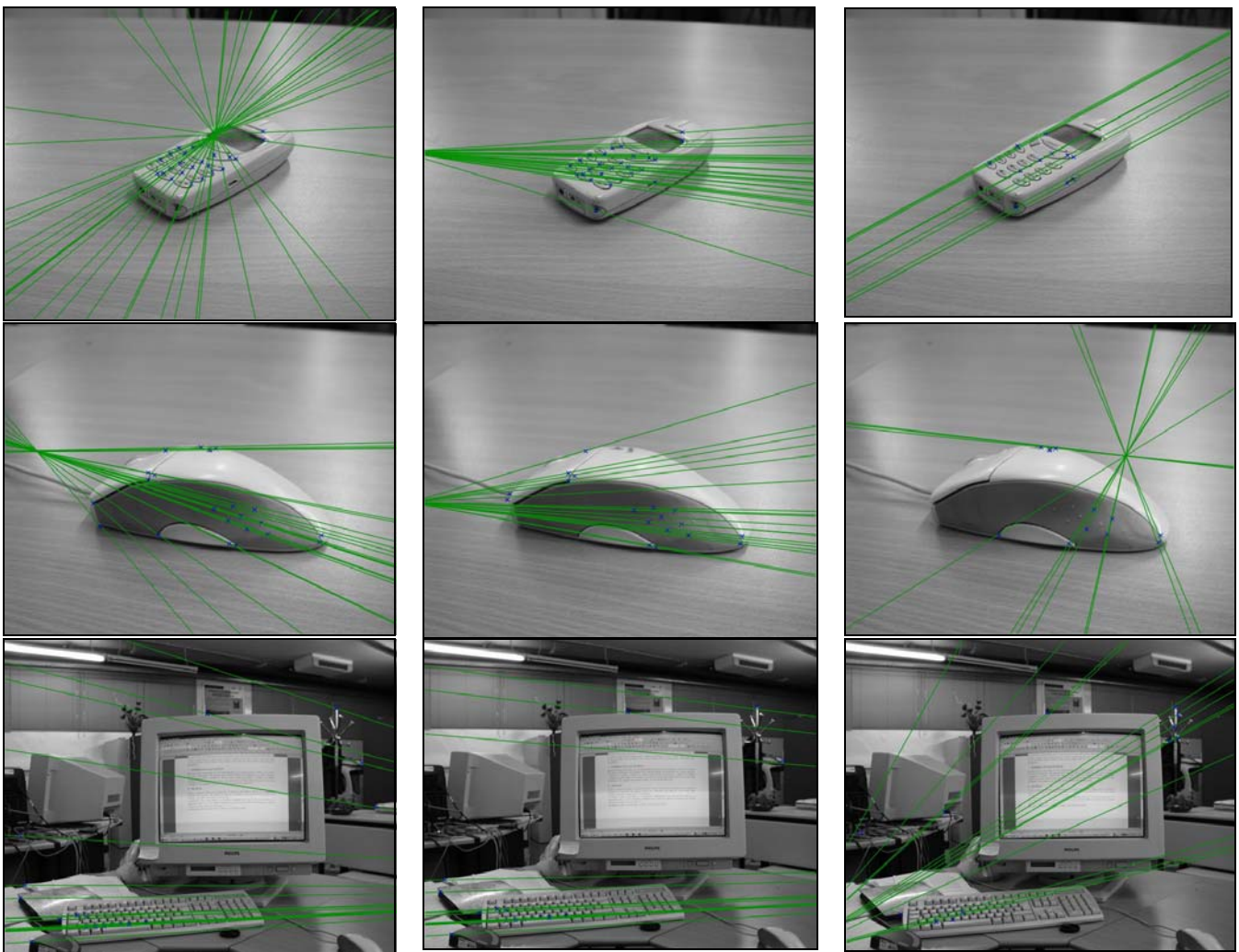


Fig. 23 - Linhas epipolares (a verde) e *inliers* (a azul) obtidos na primeira imagem de cada par de imagens de teste; da esquerda para a direita, resultado do *programa 1* usando o algoritmo de *Ransac*, resultado do *programa 1* usando o algoritmo de *LMedS*, e resultado do *programa 2*.

Como os maus resultados obtidos pelo *programa 2 (Torr's Matlab Toolkit)* apresentados nas secções 2.2.3 e 2.3.5 faziam prever, não foi possível obter a geometria epipolar utilizando os seus métodos (visto o número de *outliers* ser substancialmente superior aos *inliers*).

Dos restantes *programas*, o que aparenta melhores resultados é o *programa 3 (OpenCV)*, embora se note a necessidade de obter pontos de emparelhamento mais adequados no sentido de se conseguir melhores resultados na estimativa da geometria epipolar.

Após o cálculo da geometria epipolar, a plataforma apresenta a matriz fundamental obtida, Fig. 24.

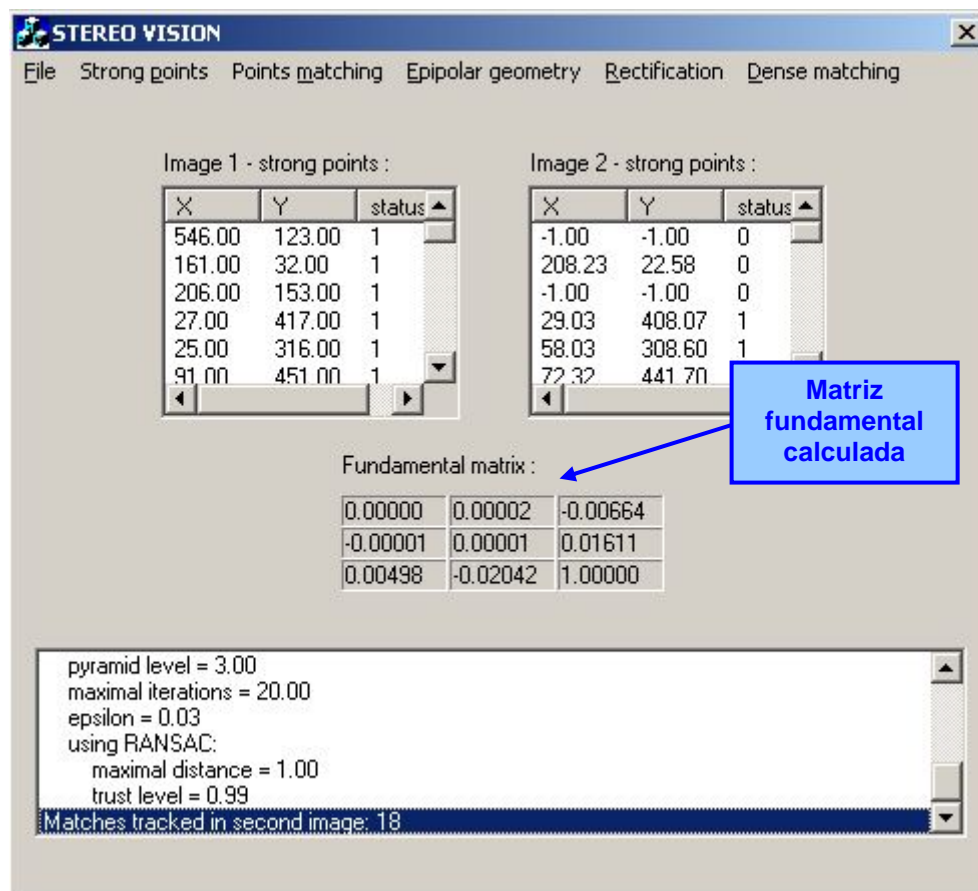


Fig. 24 - Interface da plataforma computacional após a determinação da geometria epipolar.

2.5. Rectificação

Se a geometria epipolar, traduzida pela matriz fundamental, é conhecida, então é possível restringir o problema da correspondência de pontos entre duas imagens usando as linhas epipolares, utilizando a equação (10). Melhor ainda, se as imagens forem alteradas de forma a ficarem coplanares a um plano comum, as linhas epipolares tornam-se paralelas ao

eixo horizontal da imagem: operação de rectificação, Fig. 25. Desta forma, torna-se mais fácil a resolução do problema posterior de emparelhamento denso entre imagens.

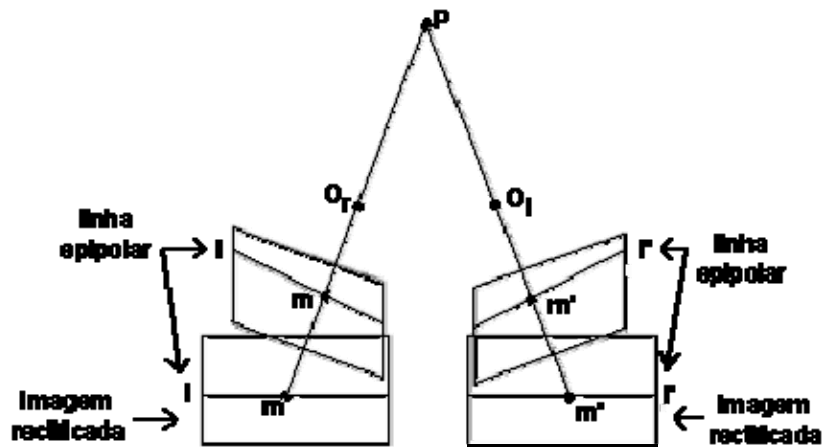


Fig. 25 – Rectificação de duas imagens estéreo.

2.5.1. Rectificação projectiva sem cálculo da geometria epipolar

Apresentado em 1999 por *Francesco Isgrò e Emanuele Trucco* [Isgrò, 1999], este é o algoritmo utilizado pelo *programa 5*, que é o único a realizar a operação de rectificação. Como o nome indica, este algoritmo rectifica duas imagens estéreo, sem para isso ser necessário a determinação da geometria epipolar, ou mais especificamente, da matriz fundamental. Assim, ao invés de determinar os pontos epipolares, e com eles determinar duas homografias H e H' que os mapeiem no infinito, como geralmente fazem os algoritmos de rectificação mais usuais [Hartley, 1998], este algoritmo explora o facto de a matriz fundamental \bar{F} de um par de imagens rectificadas ter uma forma particular e bem conhecida:

$$\bar{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (17)$$

Assim, obtidas as homografias de rectificação H e H' , é possível calcular a matriz fundamental F , utilizando a equação:

$$F = H'^T \bar{F} H. \quad (18).$$

Combinando as equações (9) e (18), verifica-se que é possível obter H e H' directamente dos pontos de correspondência entre duas imagens, mesmo na presença de ruído e de *outliers*, minimizando a função de custo \mathcal{F} :

$$\mathcal{F}(H, H') = \sum_{i=1}^N [(H'm'_i)^T \bar{F} H m_i]^2. \quad (19)$$

A minimização da equação (19), é um problema não-linear de minimização de mínimos quadrados, possível de resolver, por exemplo, com o algoritmo iterativo de *Levenberg-Marquardt* ([Press, 1992]), que necessita de uma boa estimativa inicial de H_0 e H'_0 para garantir convergência. Tal é obtido minimizando:

$$\mathcal{F}_{\rho}(H, H') = \sum_{i=1}^N \rho((H'm'_i)^T \bar{F} H m_i), \quad (20)$$

em que $\rho(z) = \log(1+1/2z)$ tem uma distribuição *Lorentziana* em imagens de baixa resolução.

A partir de H_0 e H'_0 , filtra-se os *outliers* rejeitando aqueles cujo resíduo r_i é superior a $5.2 \times med_i \{|r_i - med_j r_j|\}$, com:

$$r_i = |(H'_0 m'_i)^T \bar{F} H_0 m_i|. \quad (21)$$

Obtidos H_0 e H'_0 , itera-se a equação (19) em imagens de resolução cada vez mais elevada, até ser obtido H e H' , a partir das quais \bar{F} é calculada pela equação (18).

2.5.2. Resultados obtidos

Como o *programa 5 (Projective Rectification without Epipolar Geometry)* necessita de alguns pontos de emparelhamento, para cada par de imagens estéreo experimentais, foi escolhido o método de emparelhamento de pontos fortes que obtivesse os melhores resultados.

Os resultados obtidos estão apresentados na Fig. 26. Como se pode observar, ambas as imagens do par estéreo, sofrem uma distorção, que é proporcional à qualidade dos resultados obtidos no cálculo da geometria epipolar.

Não foi possível proceder à rectificação do par de imagens *Rato_1* e *Rato_2*, visto que as matrizes de homografia H e H' da equação (18) determinadas pelo *programa 5* eram nulas.

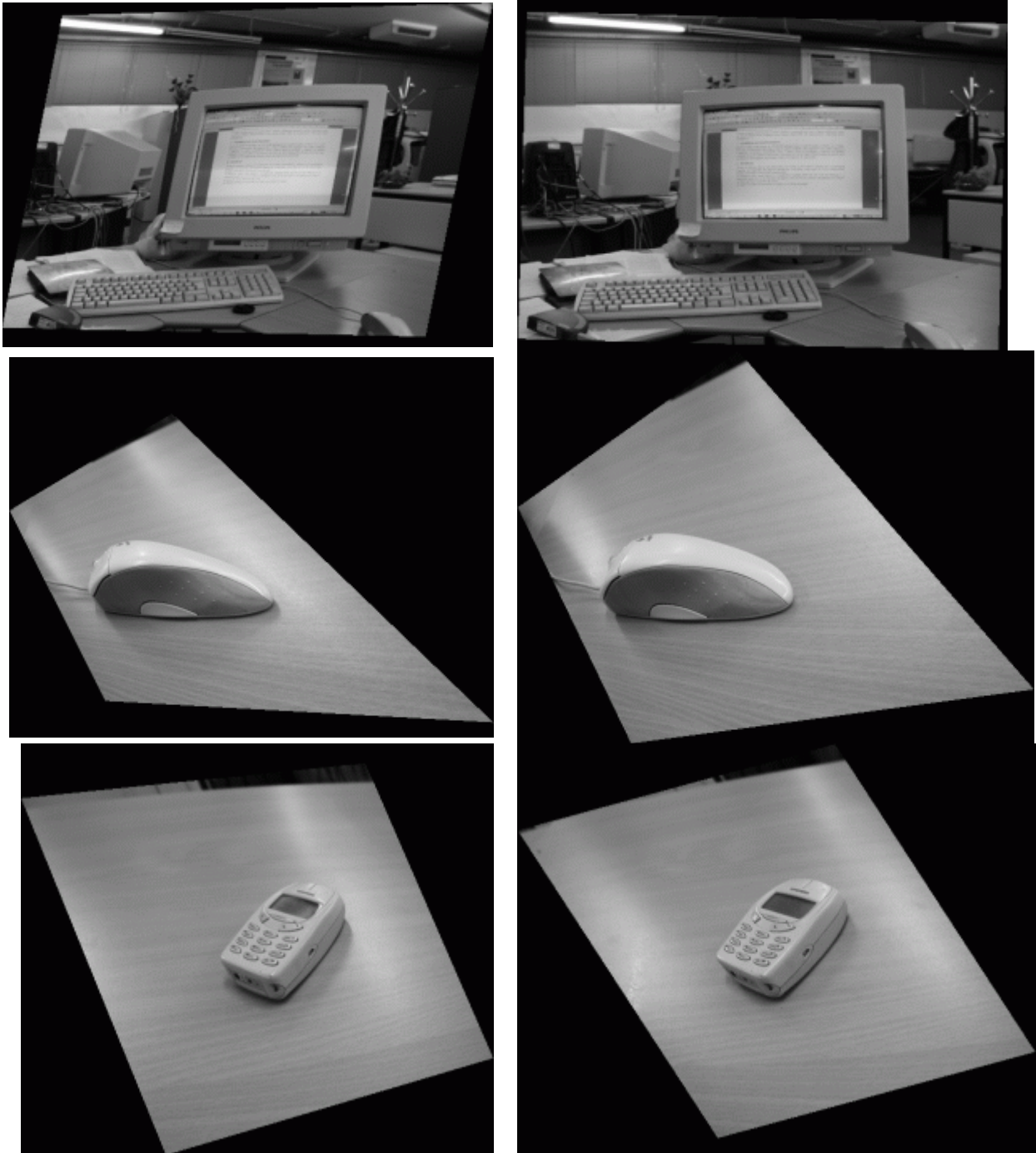


Fig. 26 - Resultado da rectificação das imagens de teste pelo *programa 5*.

2.6. Mapa de disparidade

Num sistema estéreo, a informação 3D de uma cena é representada pela disparidade entre duas imagens. Assim, o mapa de disparidade codifica a distância dos objectos em relação à(s) câmara(s), ou seja, pontos muito distantes têm disparidade zero (geralmente 0, equivalente ao preto) e pontos muito próximos terão a máxima disparidade (geralmente 255,

correspondente ao branco). Em resumo, um mapa de disparidade dá a percepção das discontinuidades em termos de profundidade de uma cena, Fig. 27.

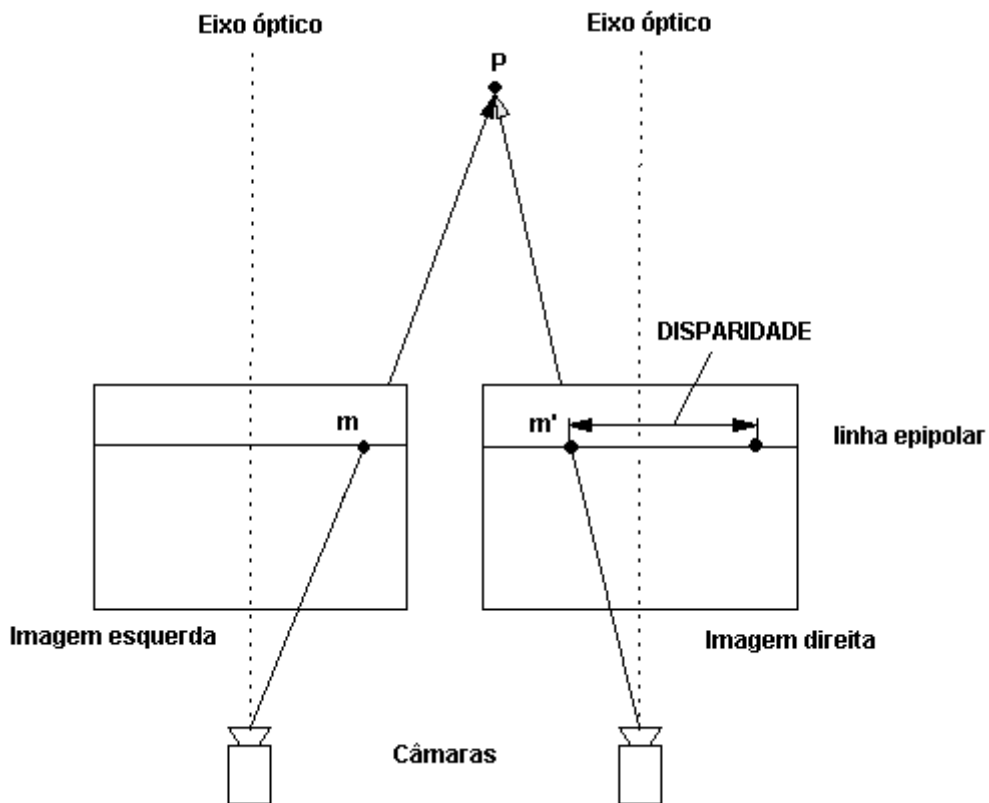


Fig. 27 - Disparidade entre dois pontos m e m' (localizados em imagens previamente rectificadas).

Somente os *programas 3 e 6*, realizam esta operação, Fig. 28, utilizando o algoritmo de *Stan Birchfield* (descrito na secção seguinte), com a diferença de que o *programa 6* também retorna um mapa de discontinuidades definido pelos píxeis que fazem fronteira entre uma mudança de pelo menos dois níveis de disparidade.

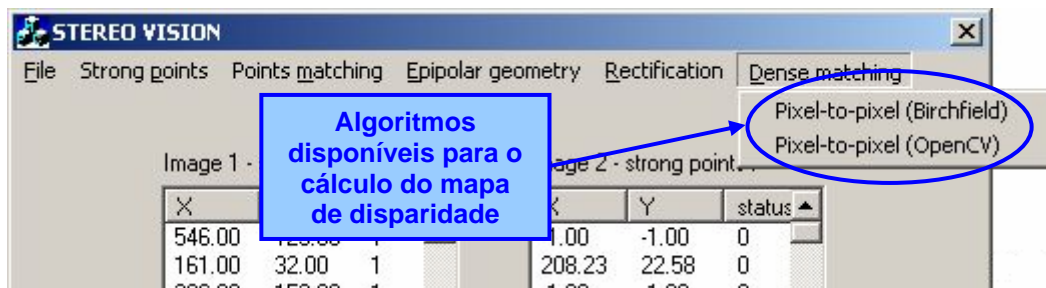


Fig. 28 - Algoritmos disponíveis, na plataforma computacional, para a determinação do mapa de disparidade.

2.6.1. Algoritmo de *Stan Birchfield*

Desenvolvido em 1999 [Birchfield, 1999], este algoritmo toma como ponto de partida duas imagens rectificadas. Primeiro, este algoritmo realiza um processo de emparelhamento denso, linha a linha na horizontal (daí a necessidade das imagens serem previamente rectificadas): o algoritmo mede a dissimilaridade entre os pontos, atribuindo a cada sequência de correspondências M um custo $\gamma(M)$, que traduz o quão improvável é que essa sequência seja uma correspondência correcta:

$$\gamma(M) = N_{occ}k_{occ} - N_mk_r + \sum_{i=1}^{N_m} d(m_i, m'_i). \quad (22)$$

Nesta equação, os parâmetros N_{occ} e N_m , representam o número de oclusões e de emparelhamentos, sendo k_{occ} e k_r as respectivas constantes de “penalidade” e “recompensa”, e $d(m_i, m'_i)$ a dissimilaridade entre os píxeis m_i e m'_i :

$$d(m_i, m'_i) = \min \{ \bar{d}(m_i, m'_i, I, I'), \bar{d}(m'_i, m_i, I', I) \}, \quad (23)$$

em que \bar{d} é uma interpolação linear da intensidade da imagem entre dois píxeis [S. Birchfield, 1998].

Partindo do pressuposto de que descontinuidades são acompanhadas de variações na intensidade da imagem, são definidas restrições a cada sequência de correspondências M , tal como impor um limite máximo ao valor de d .

Como a intensidade entre linhas de uma imagem não são independentes, também se realiza uma análise coluna a coluna no mapa de disparidades. Primeiro, a disparidade de um píxel, cujos vizinhos verticais tenham disparidade igual mas diferente dele próprio, torna-se igual à dos seus vizinhos. A seguir, para ambas as direcções, horizontal e vertical, utiliza-se o gradiente na direcção a estudar da imagem original para propagar regiões de confiança no mapa de disparidade. A confiança, para cada píxel, define-se como o número de píxeis contíguos na direcção a estudar com disparidade igual à sua. Posteriormente, o mapa de disparidade é filtrado com um filtro de média, nas duas direcções sequencialmente, de forma a preservar os vértices dos objectos.

2.6.2. Resultados obtidos

Esta etapa foi testada com imagens, previamente rectificadas e indicadas pelo *programa 6 (Depth Discontinuities by Pixel-to-Pixel Stereo)*, apresentadas na Fig. 29. Os resultados obtidos estão visíveis nas Fig. 30 e Fig. 31.



Fig. 29 - Imagens originais utilizadas pelos *programas 3 e 6* para realizar o emparelhamento denso.

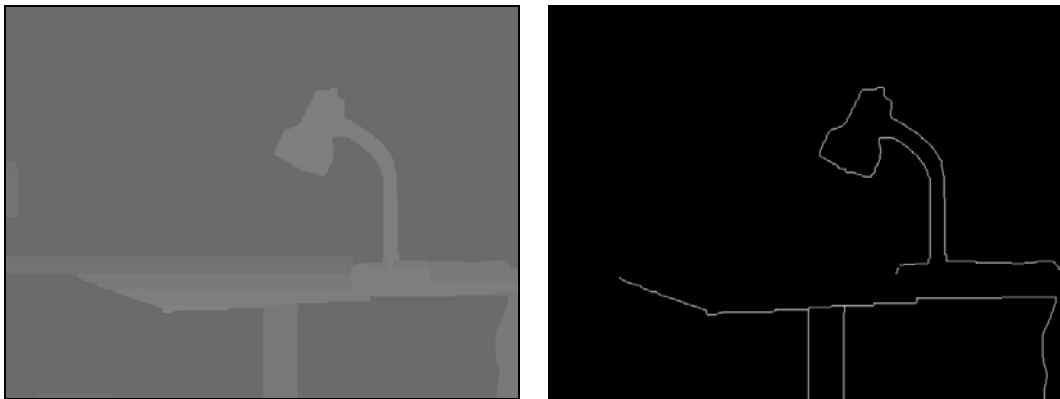


Fig. 30 - Mapas de disparidade (esquerda) e descontinuidade (direita) obtidos pelo *programa 6*.



Fig. 31 - Mapa de disparidade obtido pelo *programa 3*.

Como se pode observar nas Fig. 30 e Fig. 31, tanto o *programa 3 (OpenCV)* como o *programa 6 (Depth Discontinuities by Pixel-to-Pixel Stereo)*, obtêm bons resultados na determinação do mapa de disparidade, dadas duas imagens estéreo previamente rectificadas de forma adequada.

Capítulo 3. Conclusões e Trabalho Futuro

3.1. Conclusões

Ao longo deste trabalho, iniciou-se o desenvolvimento de uma plataforma computacional, com uma interface gráfica adequada e arquitectura modular, para permitir a aplicabilidade e a comparação das funções de Visão 3D previamente estudadas.

Em resumo, pode-se afirmar que as funções, integradas na referida plataforma e experimentalmente analisadas, obtêm bons resultados experimentais quando aplicadas em objectos que apresentam características fortes. Como foi observado, os resultados de menor qualidade estão relacionados com a determinação dos pontos fortes nas imagens a considerar e do emparelhamento entre os mesmos.

Esta debilidade, das técnicas analisadas e integradas na plataforma computacional, é tanto maior, quanto menor as variações da forma dos objectos em causa. Assim, como as funções a seguir consideradas para obter a reconstrução 3D utilizam estes pontos relevantes, os resultados obtidos pelas mesmas estão fortemente relacionados com a sua qualidade.

3.2. Trabalho futuro

As próximas etapas desta dissertação, irão concentrar-se no sentido de melhorar os resultados obtidos quando os objectos a reconstruir apresentam formas contínuas e suaves. Assim, a abordagem em perspectiva será a seguinte:

- consideração de técnicas de “escavação” do espaço tridimensional - *space carving* - para a reconstrução 3D do objecto (ver, por exemplo, [Kutulatos, 1998], [Sainz, 2002], [Seitz, 1997], [Montenegro, 2004]);
- os pontos característicos, a usar na definição do espaço 3D “a escavar”, serão determinados utilizando um reduzido número de marcadores nos objectos;
- integração de uma técnica de calibração da(s) câmara(s), assim como algoritmos de obtenção da pose e do movimento da(s) mesma(s); as técnicas base a considerar são as descritas em [Meng, 2000], [Zhang, 2000], [Zhang, 1996].

Posteriormente, pretende-se utilizar a plataforma computacional apresentada na reconstrução e caracterização de formas 3D anatómicas exteriores.

Bibliografia

-
- [1] *OpenCV - Open Computer Vision Library*, beta 4, <http://sourceforge.net/projects/opencvlibrary>, 2004.
- [2] S. Birchfield, *Depth Discontinuities by Pixel-to-Pixel Stereo*, International Journal of Computer Vision, vol. 35, no. 3, pp. 269-293, <http://vision.stanford.edu/~birch/p2p/>, 1999.
- [3] S. Birchfield, *KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker*, <http://www.ces.clemson.edu/~stb/klt/index.html>, 2004.
- [4] J.-Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker*, Intel Corporation, Microprocessor Research Labs, Technical Report, 1999.
- [5] M. C. Coelho, *Desenvolvimento de um Sistema de Levantamento de Forma 3D usando Visão por Computador*, Tese de Mestrado em Métodos Computacionais em Ciências e Engenharia, FCUP, Portugal, 2003.
- [6] N. D'Apuzzo, *Modeling human faces with multi-image photogrammetry*, SPIE Proceedings of Three-Dimensional Image Capture and Applications V, San Jose, USA, vol. 4661, pp. 191-197, 2002.
- [7] M. A. Fischler, R. Bolles, *RANdom SAMpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography*, Communications of the Association for Computing Machinery, vol. 24, no. 6, pp. 381-395, 1981.
- [8] G. Golub, C. V. Loan, *Matrix Computations*, John Hopkins University Press, 1983.
- [9] C. Harris, M. Stephens, *A combined corner and edge detector*, Forth Alvey Vision Conference, pp. 147-151, 1988.
- [10] R. Hartley, *In defense of the eight-point algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, pp. 580-593, 1997.
- [11] R. Hartley, *Theory and Practice of projective Rectification*, International Journal of Computer Vision, 1998.
- [12] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2004.
- [13] F. Isgrò, E. Trucco, *Projective rectification without epipolar geometry*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR99, Fort Collins (Colorado), vol. I, ISBN 0-7695-0149-4, pp. 94-99, 1999.
- [14] P. Kovesei, *MATLAB Functions for Computer Vision and Image Analysis*, <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns>, 2004.
- [15] K. N. Kutulatos, S. M. Steiz, *A Theory of Shape by Space Carving*, Technical Report TR692, Computer Science Department, University of Rochester, New York, USA, 1998.
- [16] B. D. Lucas, T. Kanade, *An Iterative Image Registration Technique with an Application to Stereo Vision*, International Joint Conference on Artificial Intelligence, pp. 674-679, 1981.
- [17] X. Meng, H. Li, Z. Hu, *A New Easy Camera Calibration Technique Based on Circular Points*, Proceedings of the British Machine Vision Conference, University of Bristol, UK, pp. 496-501, 2000.
- [18] A. Montenegro, P. Carvalho, M. Gattass, L. Velho, *Space Carving with a Hand-Held Camera*, Proceedings of the SIBGRAP2004 - International Symposium on Computer Graphics, Image Processing and Vision, Curitiba, Brasil, 2004.
- [19] A. Noble, *Description of image surfaces*, PhD thesis, Department of Engineering Science, Oxford University, pp. 45, 1989.
- [20] D. Parks, J.-P. Gravel, "Corner Detectors", <http://www.cim.mcgill.ca/~dparks/>
-

-
- [21] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, *Visual Modeling with a Hand-Held Camera*, International Journal of Computer Vision, vol. 59, no. 3, pp. 207-232, 2004.
- [22] M. Pollefeys, R. Koch, M. Vergauwen, L. V. Gool, *Flexible acquisition of 3D structure from motion*, Proceedings of the IEEE Workshop on Image and Multidimensional Digital Signal Processing, Alpbach, Austria, pp. 195-198, 1998.
- [23] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical recipes in C: the art of scientific computing*, Cambridge University Press, 2nd edition, 1992.
- [24] F. Remondino, *3D Reconstruction of Static Human Body with a Digital Camera*, Videometrics VII, SPIE Electronic Imaging, Santa Clara (California), USA, vol. 5013, pp. 38-45, 2003.
- [25] L. G. Roberts, *Machine Perception of 3-D Solids*, PhD Thesis, MIT Lincoln Laboratory, 1963.
- [26] P. Rousseeuw, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.
- [27] P. J. Rousseeuw, *Least Median of Squares Regression*, Journal of the American Statistical Association, vol. 79, no. 388, pp. 871-880, 1984.
- [28] C. T. S. Birchfield, *A Pixel Dissimilarity Measure that is Insensitive To Image Sampling*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 4, pp. 401-406, 1998.
- [29] M. Sainz, N. Bagherzadeh, A. Susin, *Carving 3D Models from Uncalibrated Views*, Proceedings of the 5th IASTED International Conference Computer Graphics and Imaging, Hawaii, USA, pp. 144-149, 2002.
- [30] S. N. Seitz, C. R. Dyer, *Photorealistic Scene Reconstruction by Voxel Coloring*, Proc. Computer Vision and Pattern Recognition Conference, pp. 1067-1073, 1997.
- [31] W. Shoaff, *A Short History of Computer Graphics*, Florida Institute of Technology, Department of Computer Sciences, <http://www.cs.fit.edu/~wds/classes/graphics/History/history/history.html>, 2000.
- [32] C. Tomasi, T. Kanade, *Detection and Tracking of Point Features*, Carnegie Mellon University, Technical Report CMU-CS-91-132, 1991.
- [33] P. H. S. Torr, *Bayesian model estimation and selection for epipolar geometry and generic manifold fitting*, International Journal of Computer Vision, vol. 50, no. 1, pp. 35-61, 2002.
- [34] P. H. S. Torr, *Bayesian Model Estimation and Selection for Epipolar Geometry and Generic Manifold Fitting*, International Journal of Computer Vision, 50(1), pp. 27-45, 2002.
- [35] P. H. S. Torr, *A Structure and Motion Toolkit in Matlab*, <http://wwwcms.brookes.ac.uk/~philiptorr/Beta/torrsam.zip>, 2002.
- [36] S. Ullman, *The Interpretation of Visual Motion*, Massachusetts MIT Press, Cambridge, USA, 1979.
- [37] Z. Zhang, *A New Multistage Approach to Motion and Structure Estimation: From Essential Parameters to Euclidean Motion Via Fundamental Matrix*, Research Report, no.2910, INRIA Sophia-Antipolis, 1996.
- [38] Z. Zhang, *A Flexible New Technique for Camera Calibration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, 2000.
- [39] J. Y. Zheng, *Acquiring 3D models from sequences of contours*, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 16, no. 2, pp. 163-178, 1994.
-