

VISÃO COMPUTACIONAL

AULA 6

CARACTERÍSTICAS DE IMAGENS: PONTOS E SUPERFÍCIES

Cantos (corners)

CARACTERÍSTICAS DE IMAGENS: PONTOS E SUPERFÍCIES

CARACTERÍSTICAS PONTUAIS : CANTOS (CORNERS)

- Bordas (Edges)
 - Matemática Complicada
 - Podem, também, ser caracterizadas geométrica e intuitivamente
 - Projeção de fronteiras de objetos
 - Marcos superficiais
 - Outros elementos interessantes de uma cena

- Cantos (Corners)
 - Mais fácil de caracterizar matematicamente.
 - Não correspondem necessariamente a qualquer entidade geométrica da cena.
 - Não são apenas intersecções de linhas: são também estruturas com padrões de intensidade.
 - São estáveis através de seqüências de imagens.
 - Interessante para rastreamento de objetos através de seqüências.

A estrutura de um canto pode ser caracterizada através da matriz C , na vizinhança Q de um ponto p da imagem. (tensor de estrutura, matriz de autocorrelação ou matriz de segundo momento)

$$C = \begin{bmatrix} \sum E_x^2 & \sum E_x \cdot E_y \\ \sum E_x \cdot E_y & \sum E_y^2 \end{bmatrix} \quad \text{estrutura da intensidade da imagem}$$

$$[E_x, E_y]^T = \text{gradiente da imagem} ; \quad E_x = \frac{\partial E}{\partial x}$$

C é simétrica, e portanto pode ser diagonalizada: (Shi-Tomasi, 1994)

$$C = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$\lambda_1, \lambda_2 = \text{autovalores } (\lambda_1 \geq \lambda_2)$ de dois autovetores ortogonais (direções principais)

autovetores \implies representam direção da borda

autovalores \implies representam intensidade da borda

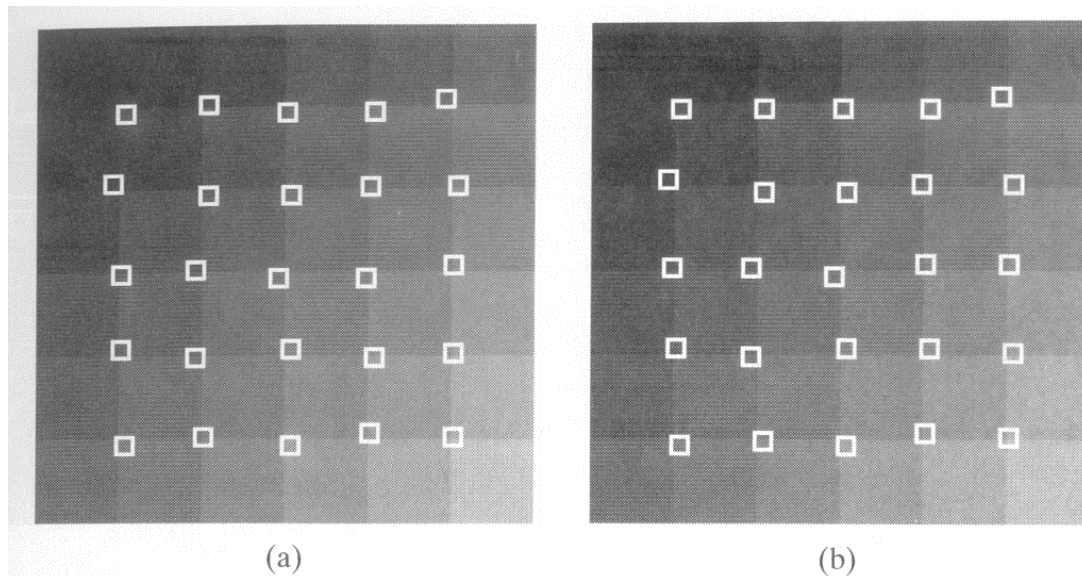
- Exemplos

Se $\lambda_1 = \lambda_2 = 0 \rightarrow Q$ perfeitamente uniforme

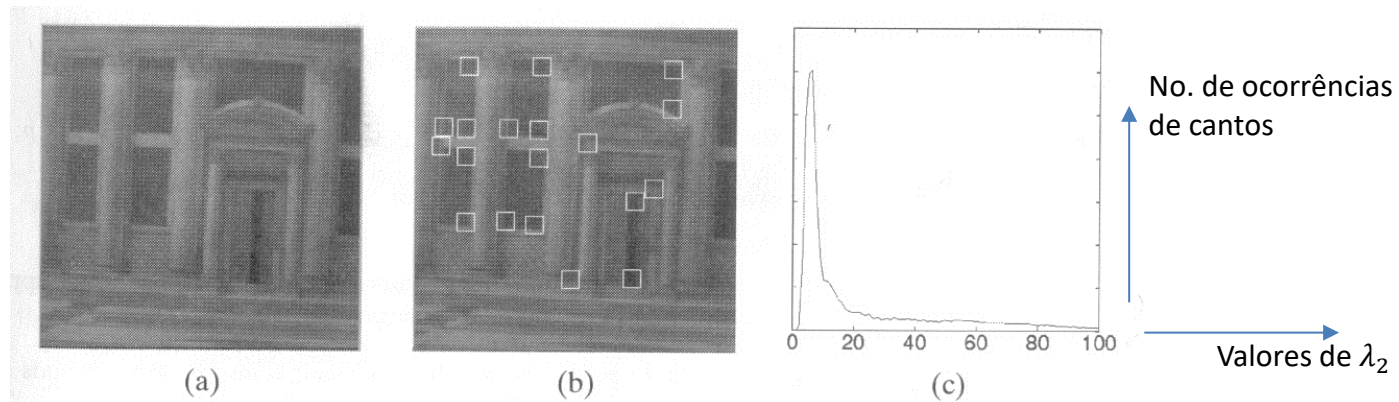
$\lambda_2 = 0 ; \lambda_1 > 0 \rightarrow Q$ contém borda de degrau ideal, branco e preto e o autovetor de λ_1 é paralelo ao gradiente da imagem

$\lambda_1 \geq \lambda_2 > 0 \rightarrow Q$ contém o canto de um quadrado preto em fundo branco (ou vice-versa).

O canto é identificado por duas bordas agudas (variação forte). Portanto, com $\lambda_1 \geq \lambda_2$, um canto é um local onde o menor autovalor, λ_2 , é grande o suficiente.

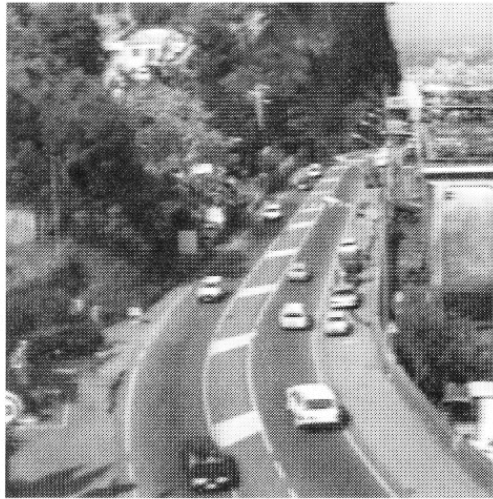


Cantos achados em uma imagem sintética de 8 bits de um tabuleiro de xadrez corrompida por 2 operações de filtragem gaussiana de $\sigma = 2$. O canto é o ponto inferior direito de cada vizinhança 15x15.

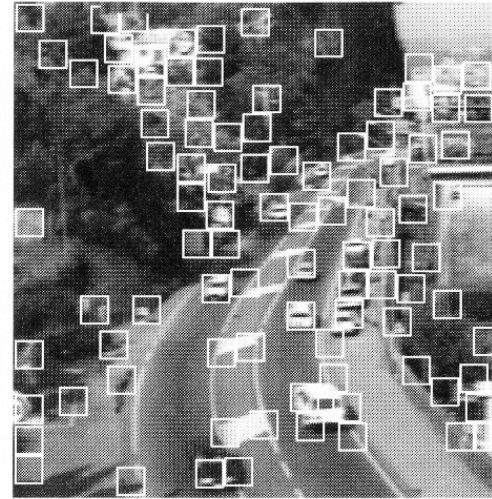


(a) imagem original de um prédio. (b): vizinhança de 15x15 pixels de alguns dos pontos da imagem para os quais $\lambda_2 > 20$ (c): histograma dos valores de λ_2 por toda a imagem.

- Os pontos característicos são cantos na imagem com alto contraste e juntas T geradas pela intersecção de contornos de objetos



(a)



(b)

(a): imagem de uma cena externa. (b): cantos achados usando uma vizinhança de 15×15 . O canto é o ponto inferior direito de cada vizinhança 15×15 .

- Incluem também cantos do padrão de intensidade local não correspondentes a características óbvias da imagem.
- Em termos gerais, nos pontos de cantos, a superfície da intensidade da imagem tem 2 direções bem pronunciadas e distintas, associadas a autovalores de C , ambos significativamente diferentes de zero.

Algoritmo CORNERS

Entrada: Imagem I

2 parâmetros: limiar de $\lambda_2 \implies \tau$

tamanho de vizinhança (janela): $2N+1$ pixels

1. Compute o gradiente da imagem sobre a imagem I inteira;
2. Para cada ponto p :
 - a) forme a matriz C sobre a vizinhança Q, $(2N+1) \times (2N+1)$, de p
 - b) compute λ_2 , o menor autovalor de C;
 - c) se $\lambda_2 > \tau$ salve as coordenadas de p dentro de uma lista, L.
3. Classifique L em ordem decrescente de λ_2 .
4. Faça varredura da lista do topo à base: para cada ponto corrente, p , apague todos os pontos aparecendo posteriormente na lista que pertençam à vizinhança de p .

A saída é uma lista de pontos característicos em que $\lambda_2 > \tau$ e cujas vizinhanças não se sobreponham.

- Os parâmetros τ e $(2N+1)$ podem ser estimados de:
 - $\tau \rightarrow$ histograma de λ_2 da imagem
 - $N \rightarrow$ de 2 a 10 para a maioria dos casos práticos

Algoritmo baseado no de Tomasi & Kanade, 1991

- O algoritmo é bem robusto

Outros Detectores Similares

(Equações diferentes para o cálculo da “força” do canto)

- Harris & Stephen (1988)

$$\det(C) - \alpha \cdot \text{traço}(C)^2 = \lambda_2 \lambda_1 - \alpha (\lambda_2 + \lambda_1)^2 \quad ; \quad \alpha \cong 0,06$$

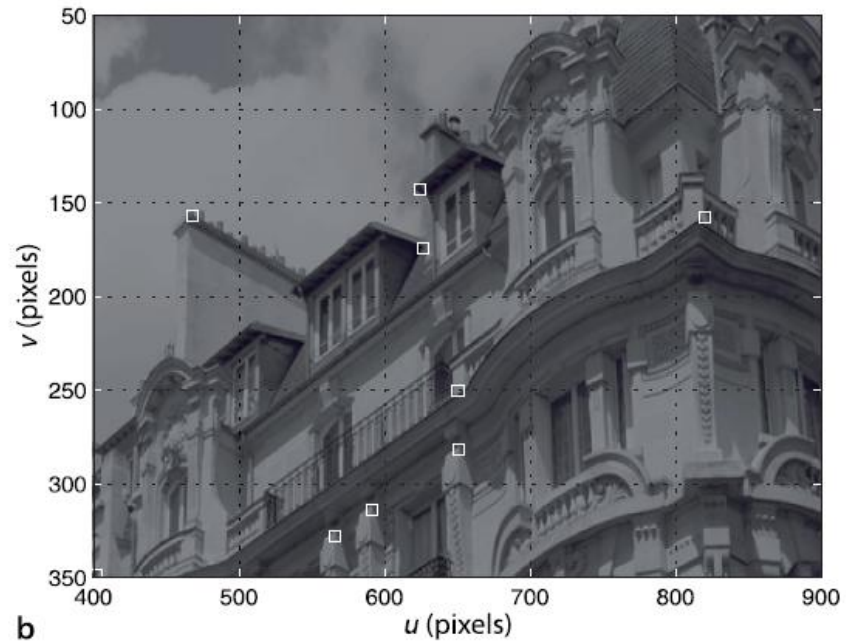
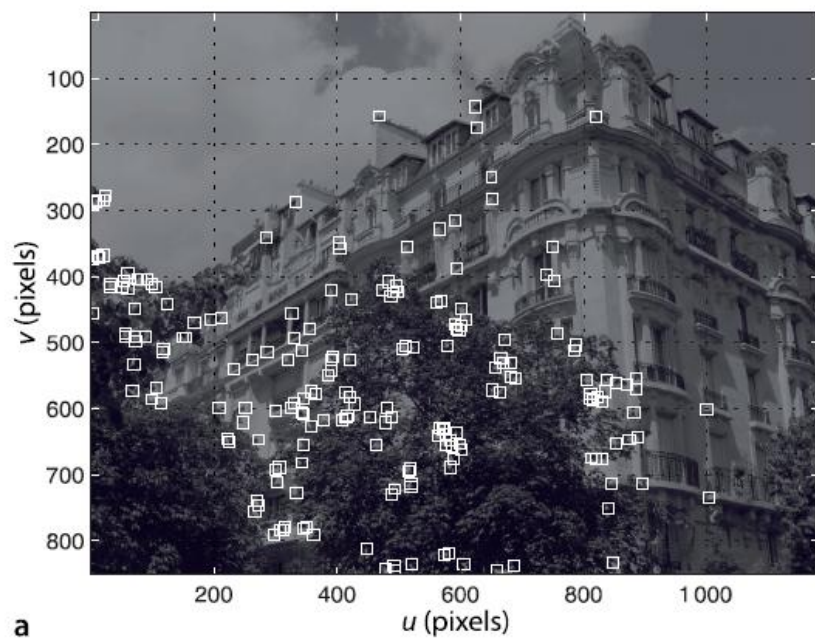
- Não exige o uso de raiz quadrada e ainda é invariante à rotação e reduz o peso de características parecidas com borda onde $\lambda_1 \gg \lambda_2$

- Triggs (2004)

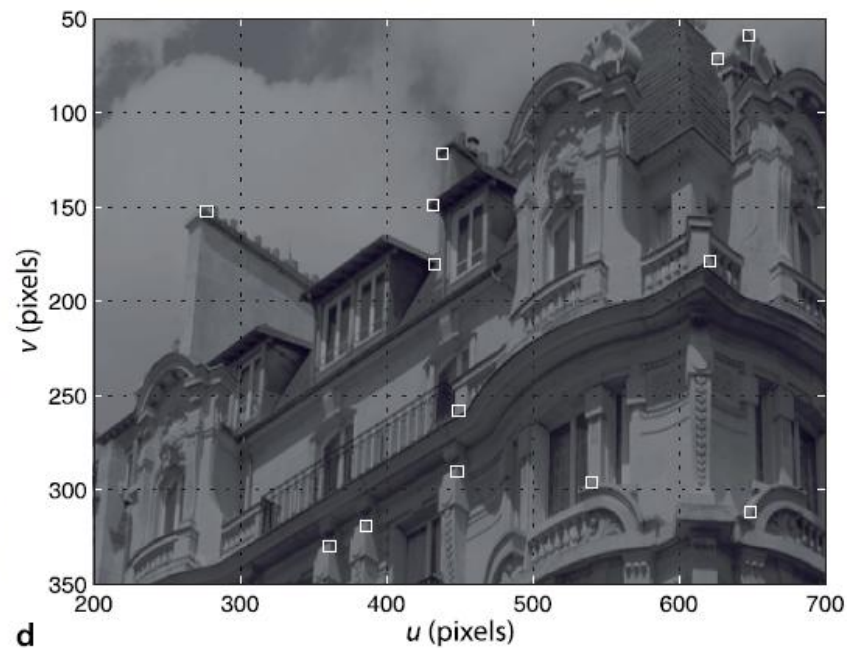
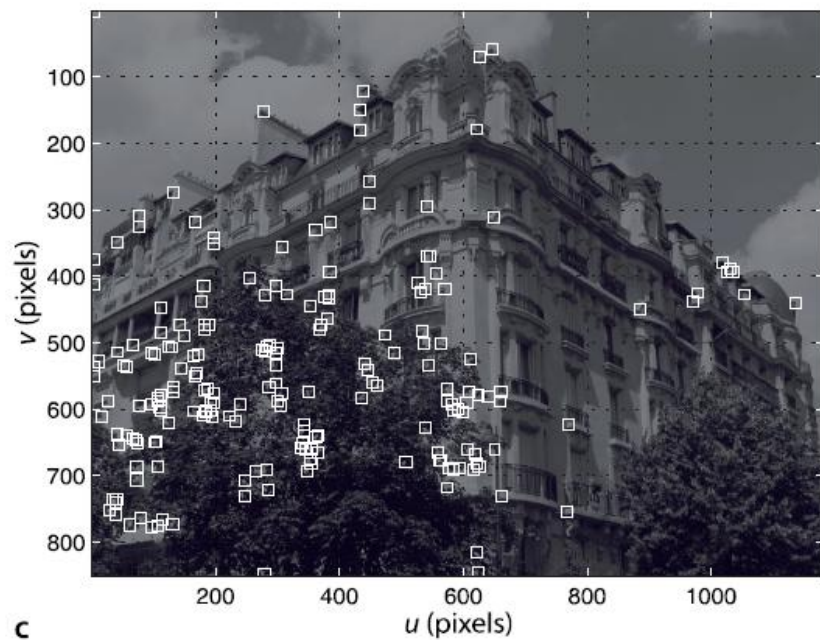
$$\lambda_2 - \alpha \cdot \lambda_1 \quad ; \quad \alpha \cong 0,05$$

- Brown, Szeliski & Winder (2005)

$$\det(C)/\text{traço}(C) = (\lambda_2 \cdot \lambda_1) / (\lambda_2 + \lambda_1)$$



Detector de Harris: a) vista 1; b) zoom da vista 1; c) vista 2; d) zoom da vista 2



Desvantagens do Detector de Harris ou similares

- **É sensível a mudanças de escala**

Se a distância entre a câmera e a cena muda, fazendo com que os objetos pareçam maiores ou menores, o detector de Harris não consegue encontrar os mesmos cantos de forma confiável. Isso é um problema em muitas aplicações do mundo real.

- **Detecta texturas finas**

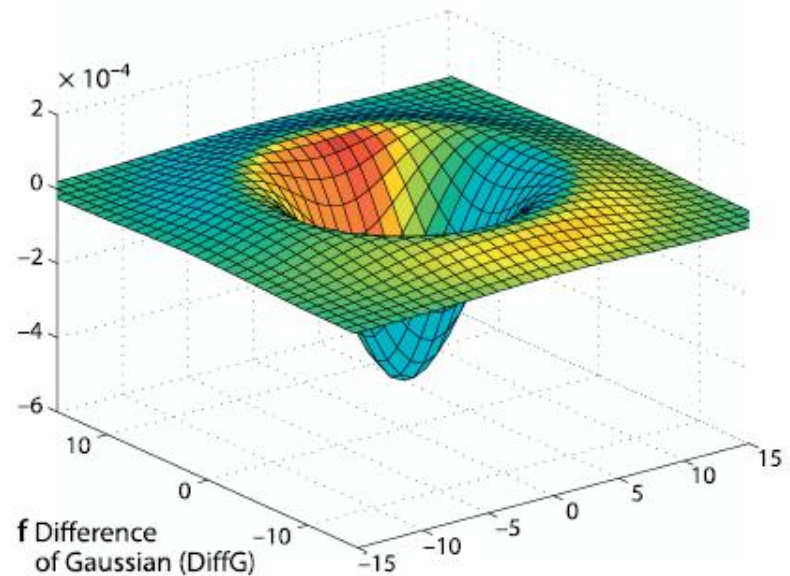
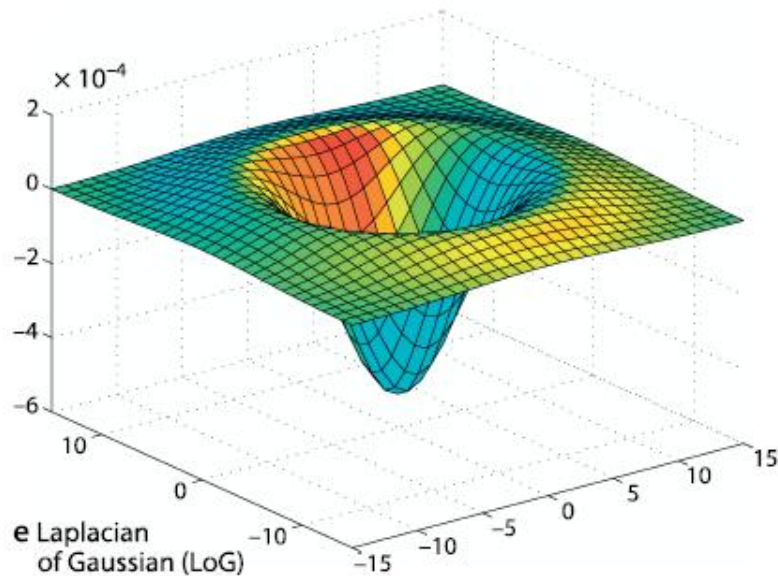
O detector é muito sensível a detalhes pequenos e repetitivos, como folhas de árvores. Ou seja, pode não detectar características que representem a estrutura maior da cena, detectando apenas o "ruído" da textura fina.

Surge a necessidade de um método de detecção de pontos de interesse que seja mais **robusto a variações de escala** e capaz de identificar características de maior importância na imagem.

Objeto mais próximo → mais detalhes → menor escala

Objeto mais distante → menos detalhes → maior escala

Detectores de Cantos no Espaço de Escalas (Scale-Space)



$$\nabla^2 I = L \otimes (G(\sigma) \otimes I) = \underbrace{(L \otimes G(\sigma))}_{\text{LoG}} \otimes I$$

Pode ser aproximado por

$$\text{DiffG}(u, v; \sigma_1, \sigma_2) = G(\sigma_1) - G(\sigma_2) = \frac{1}{2\pi\sigma_1^2\sigma_2^2} \left(\sigma_2^2 e^{-\frac{u^2+v^2}{2\sigma_1^2}} - \sigma_1^2 e^{-\frac{u^2+v^2}{2\sigma_2^2}} \right) \quad \sigma_1 > \sigma_2$$

Em geral

$$\sigma_1 = 1.6 \sigma_2$$

Detectores de Cantos no Espaço de Escalas (Scale-Space)

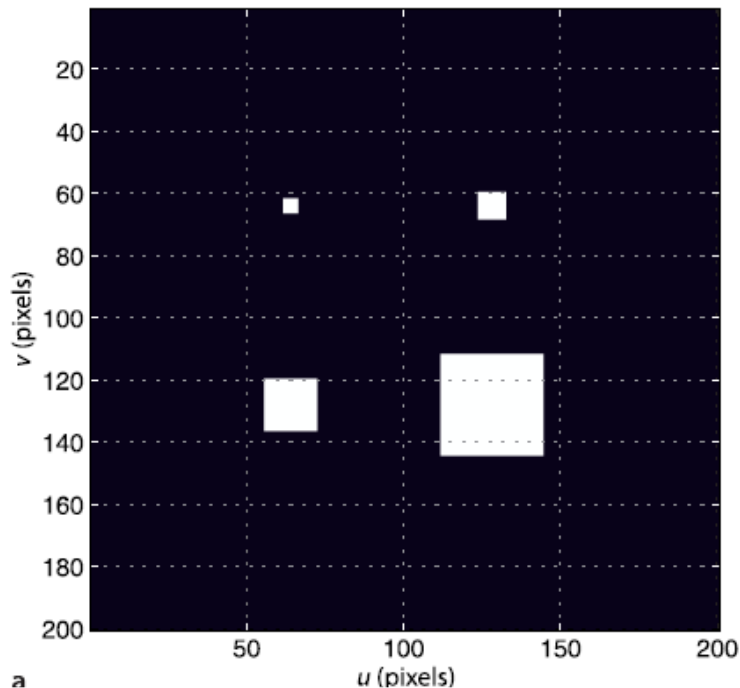
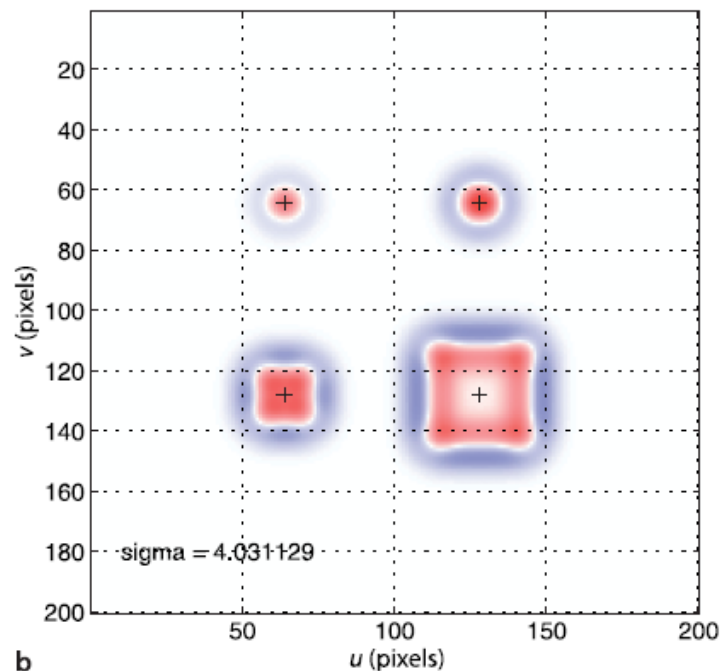


Imagem sintética, com blocos de 5x5, 9x9, 17x17 e 33x33 pixels

Quadrados vistos em diferentes escalas, como se fossem o mesmo quadrado visto de distâncias diferentes



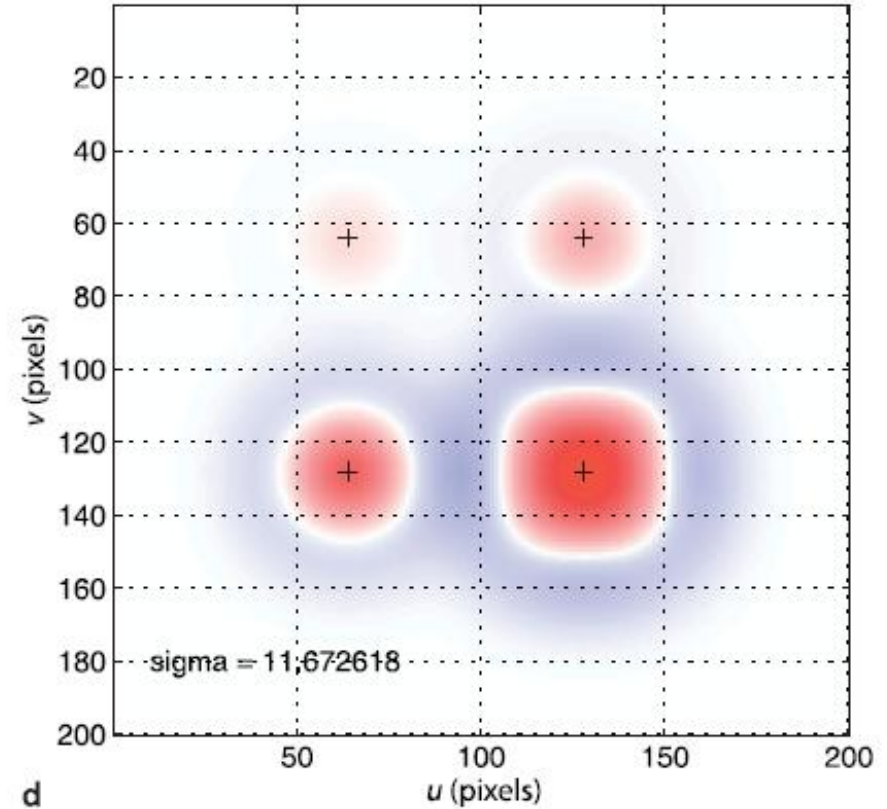
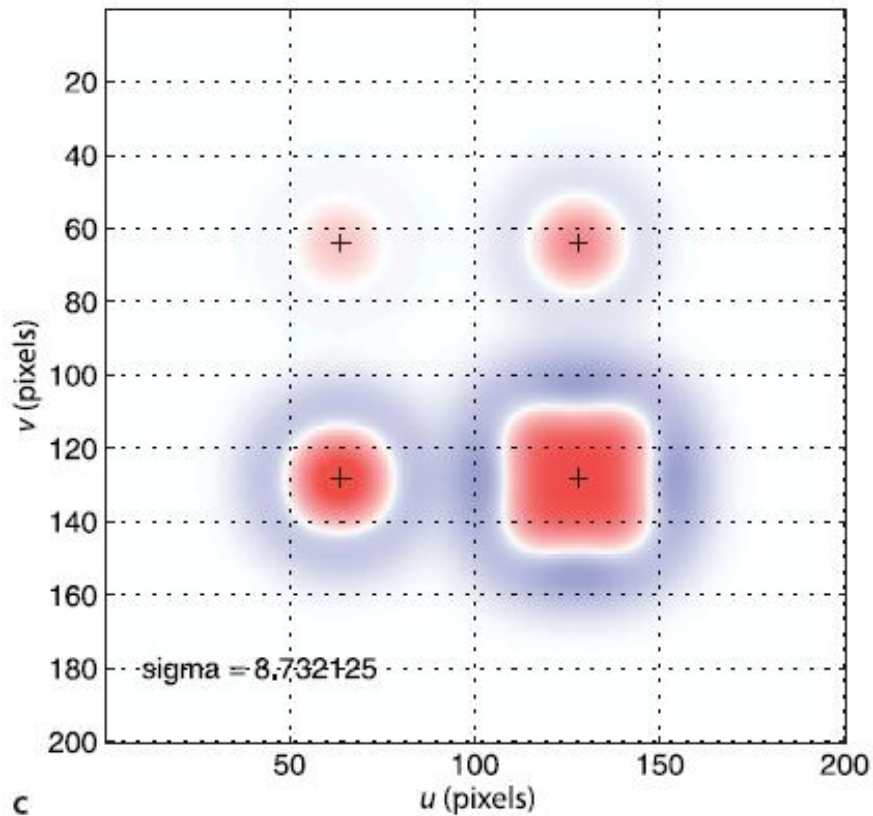
Aplicação do filtro LoG normalizado, com σ variável (desvio padrão do filtro gaussiano)

Vermelho = negativo

Azul = positivo

A escala é simulada em objetos de diferentes tamanhos: Sigma maior → maior escala

Detectores de Cantos no Espaço de Escalas (Scale-Space)

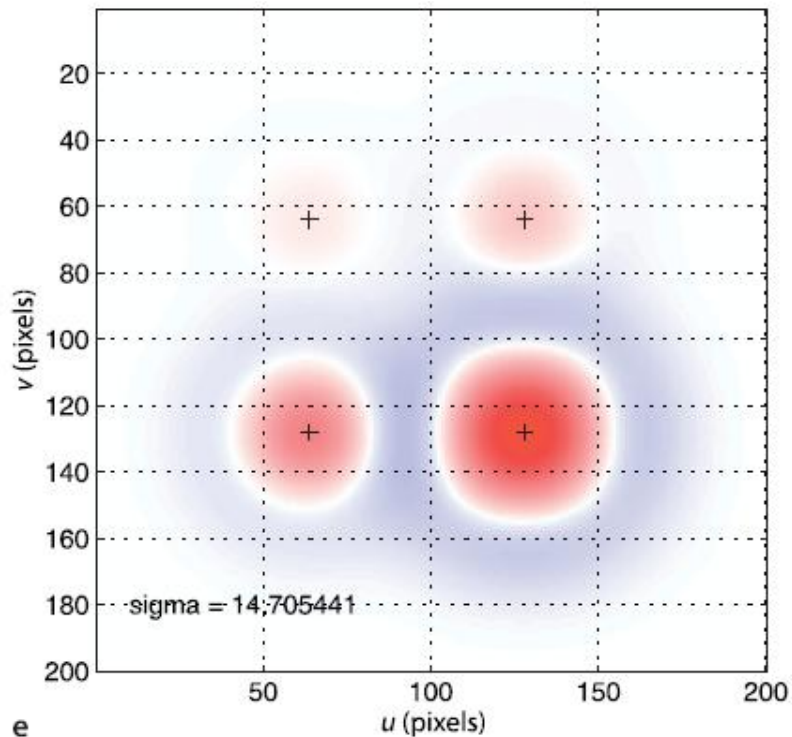


Aplicação do filtro LoG normalizado,
com σ variável

Vermelho = negativo

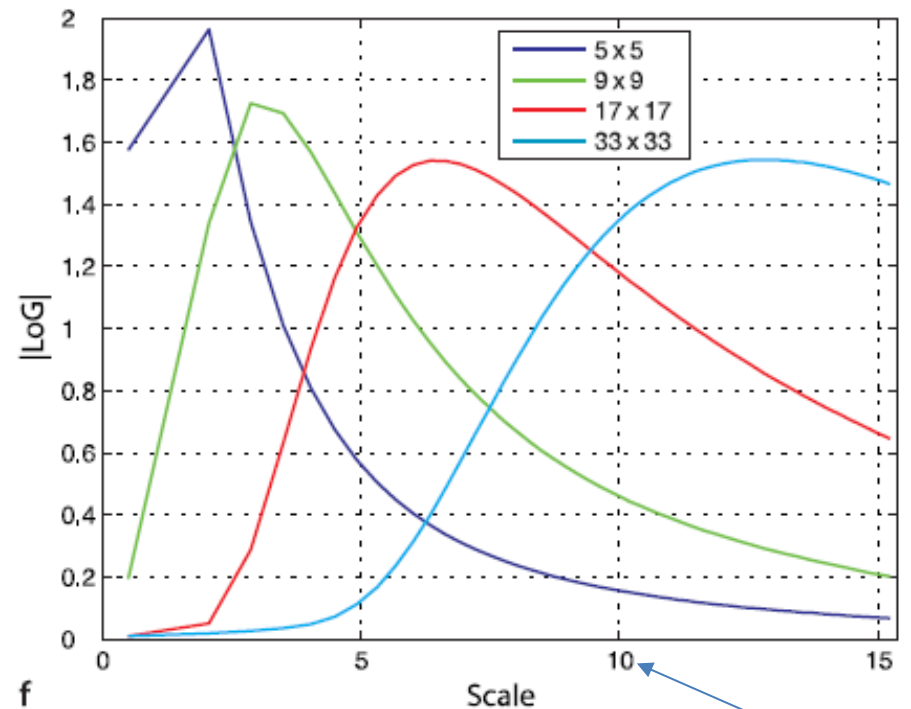
Azul = positivo

Detectores de Cantos no Espaço de Escalas (Scale-Space)



Aplicação do filtro LoG normalizado,
com *sigma* variável

Vermelho = negativo
Azul = positivo



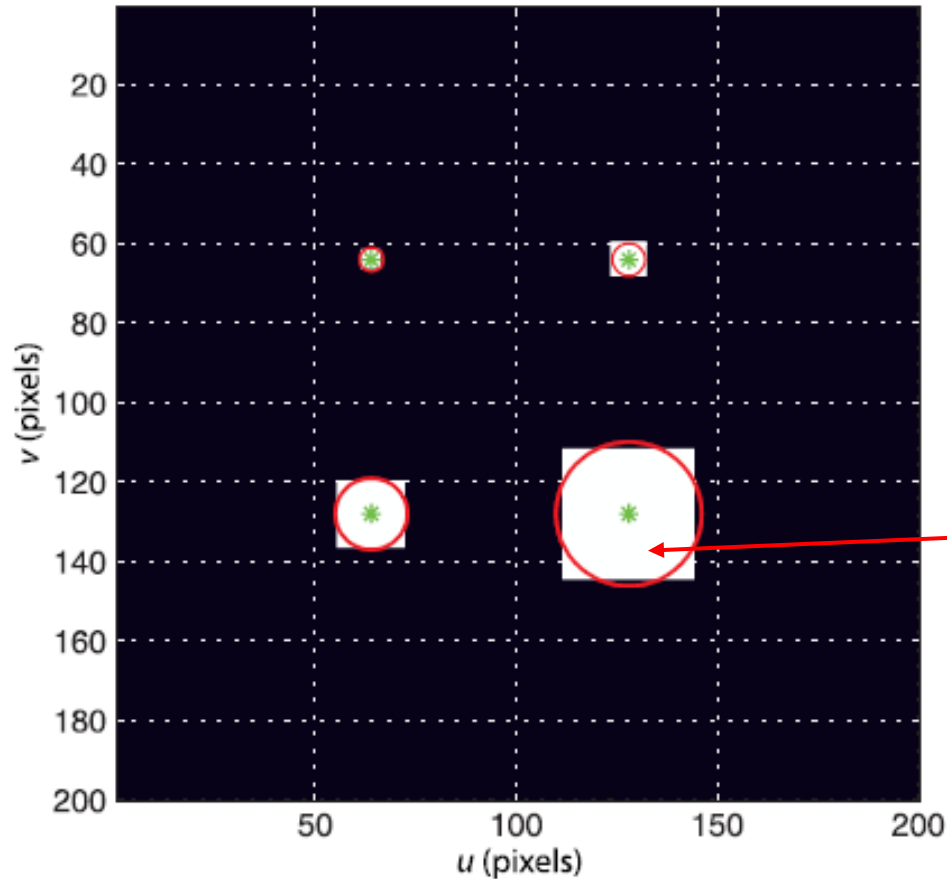
Magnitude do LoG no centro de cada
quadrado.

Cada quadrado tem um pico de
Laplaciano em uma escala diferente

Sigma

Detectores de Cantos no Espaço de Escalas (Scale-Space)

Supõe-se uma imagem 3D, com a terceira dimensão sendo a escala. Plota-se o círculo vermelho correspondente à dimensão com maior Laplaciano, na sua respectiva coordenada, conforme no gráfico

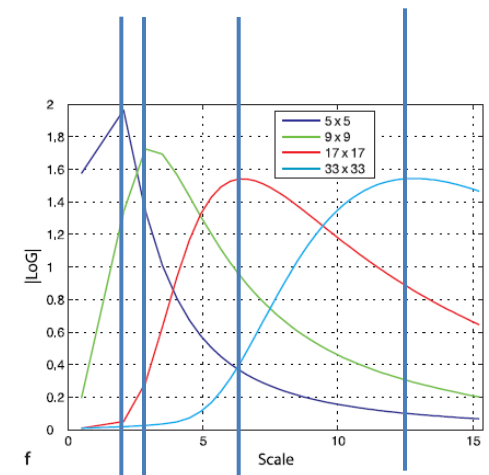


Pontos de valor máximo de LoG para cada escala

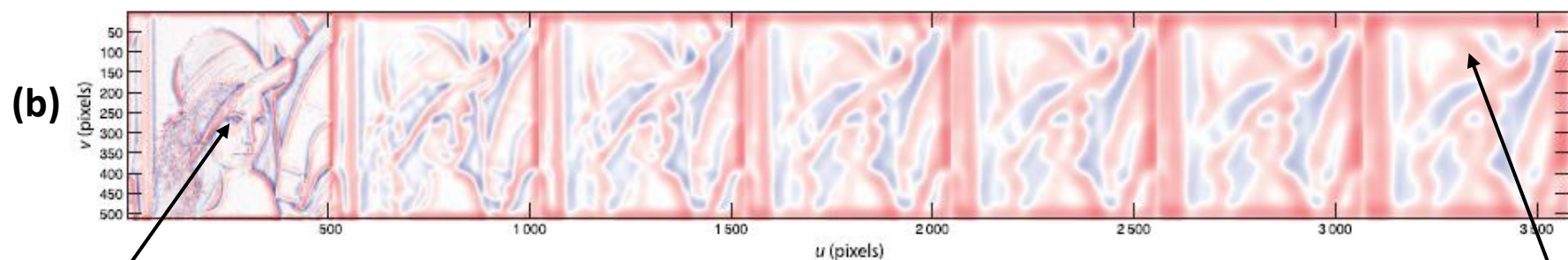
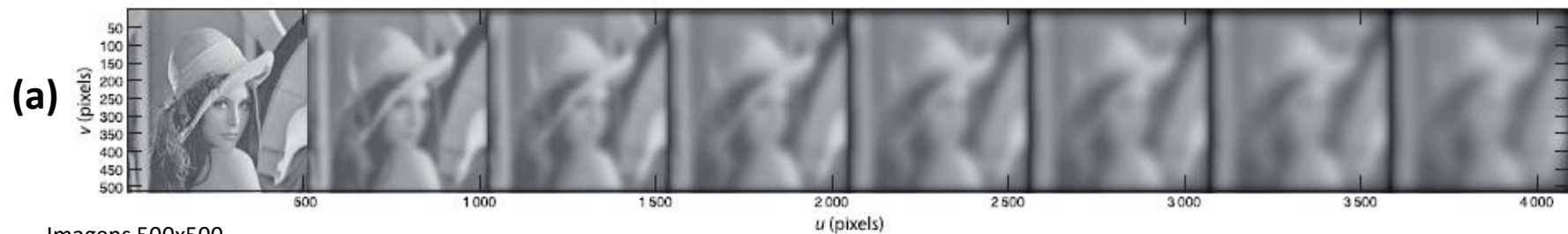
Raio do círculo indica o valor da escala

Região suporte

Imagem sintética com superposição de caractere e indicador de escala



Detectores de Cantos no Espaço de Escalas (Scale-Space)

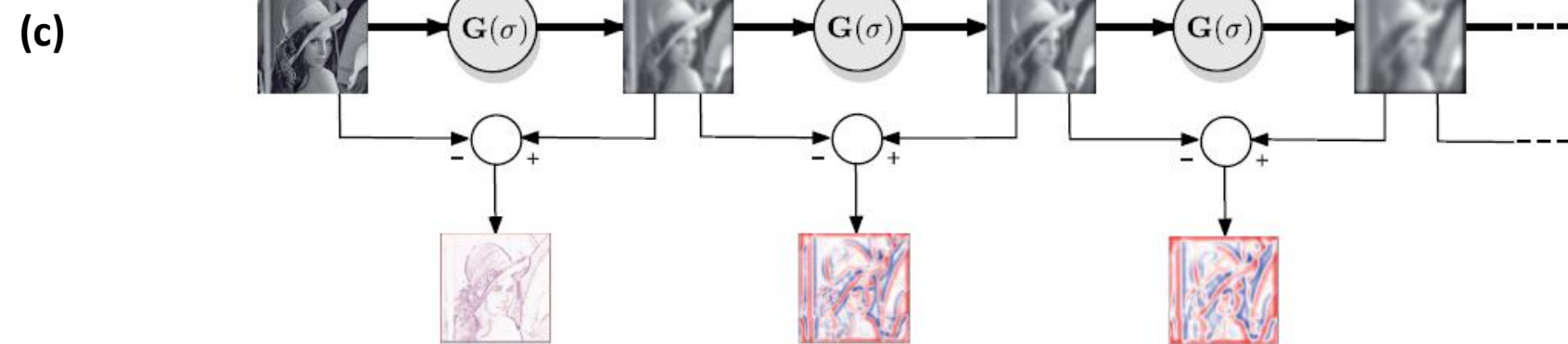


Sequência de espaço de escalas em imagem real para sigma = 2, Filtro Gaussiano (a) e LoG (b)

Gaussian sequence →

Olhos ressaltados

Chapéu com baixo gradiente



$$\text{DiffG}(u, v; \sigma_1, \sigma_2) = G(\sigma_1) - G(\sigma_2)$$

$$\sigma_1 = 1.6\sigma_2$$

Laplacian of Gaussian (LoG) sequence →

Cálculo da sequência de Gaussiano e LoG no espaço de escalas (c)

Detectores de Cantos no Espaço de Escalas (Scale-Space)

- SIFT (Scale-Invariant Feature Transform) – ou PCA (Principal Component Analysis) + SIFT
 - Baseado em cálculo de gradientes e na sequência de Diferença Gaussiana, com redução e diagonalização de janelas no PCA.

PATENTEADO

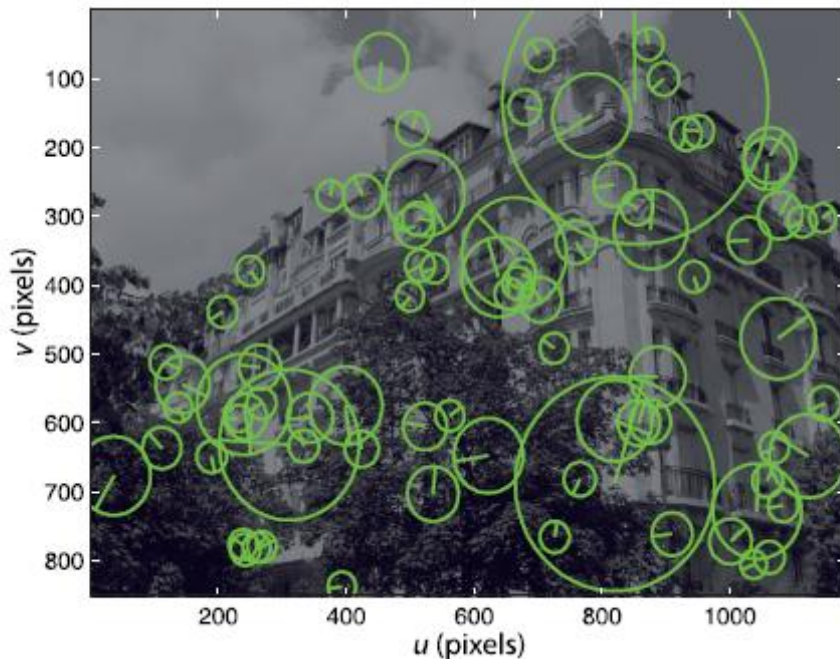


Detectores de Cantos no Espaço de Escalas (Scale-Space)

- **SIFT (Scale-Invariant Feature Transform)** – ou PCA (Principal Component Analysis) + SIFT
 - **Baseado em cálculo de gradientes e na sequência de Diferença Gaussiana, com redução e diagonalização de janelas no PCA.**
 - **Detecção de Cantos:** SIFT usa uma diferença de Gaussianas (DoG) para identificar cantos em diferentes escalas, tornando-o invariante à escala.
 - **Descritor:** Cria um descritor de 128 dimensões com base na orientação dos gradientes em torno do canto, o que o torna robusto à rotação.
 - **Vantagens:** Alta precisão e robustez a variações de escala, rotação e iluminação.
 - **Desvantagens:** Computacionalmente caro e patenteado.

Detectores de Cantos no Espaço de Escalas (Scale-Space)

- **SURF – Speeded-Up Robust Feature**



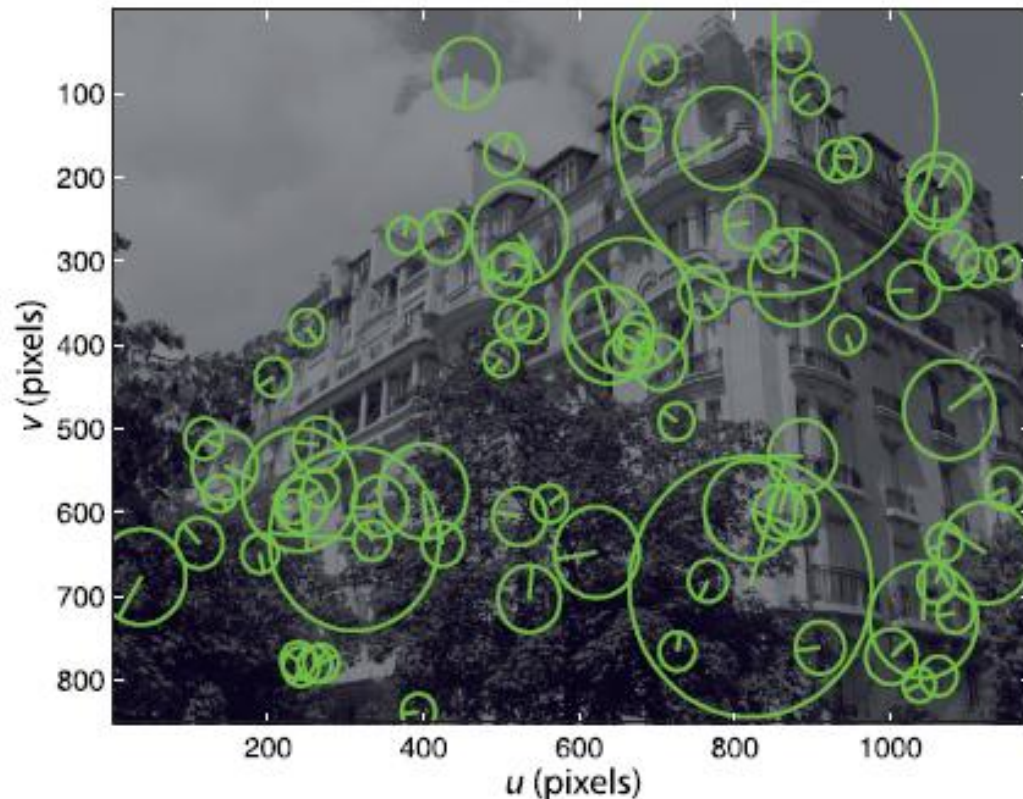
Mais rápido mas menos preciso que o SIFT

- Baseado em máximos de uma sequência aproximada de Hessian of Gaussian (matriz de duplo gradiente)
- **Robusto a variações de brilho, escala e rotação**
- **Detecção de Cantos:** SURF usa o determinante Hessiano baseado em imagens integrais para acelerar a detecção de cantos.
- **Descritor:** Utiliza um descritor baseado em Haar wavelets, que é mais rápido de calcular que o SIFT.
- **Vantagens:** **Mais rápido que o SIFT, mantendo boa precisão e robustez.**
- **Desvantagens:** **Menos preciso que o SIFT em algumas situações.**

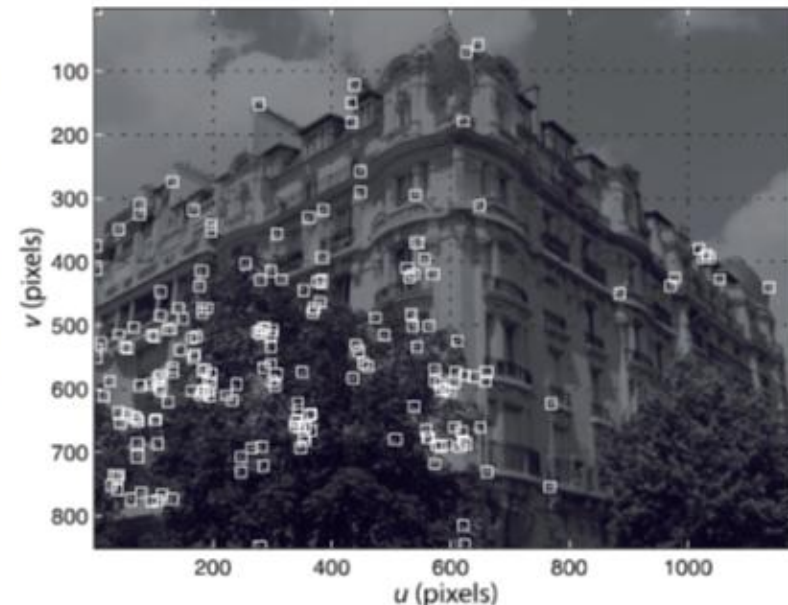
Detectores de Cantos no Espaço de Escalas (Scale-Space)

- SURF – Speeded-Up Robust Feature

Descritores SURF mostrando a região suporte (correspondendo à escala: parte interna do círculo) e a orientação (da borda dominante) como uma linha radial



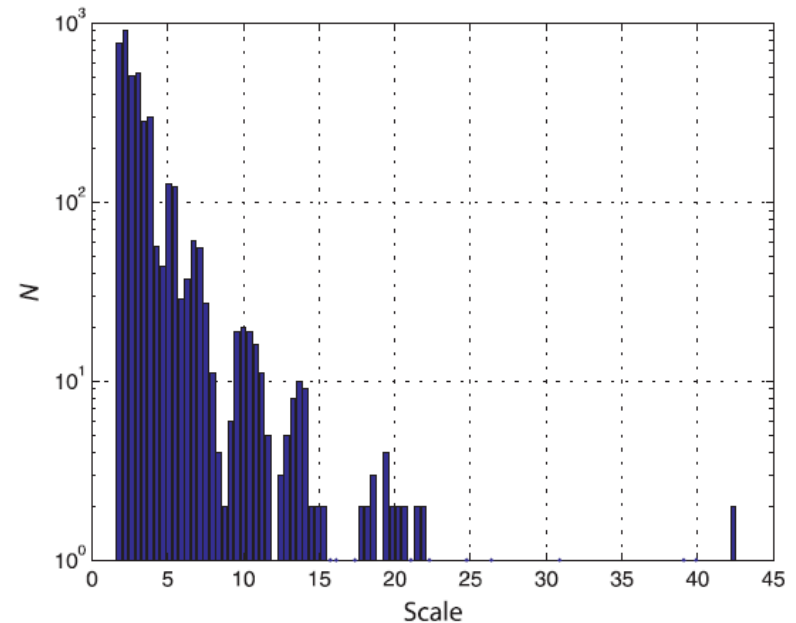
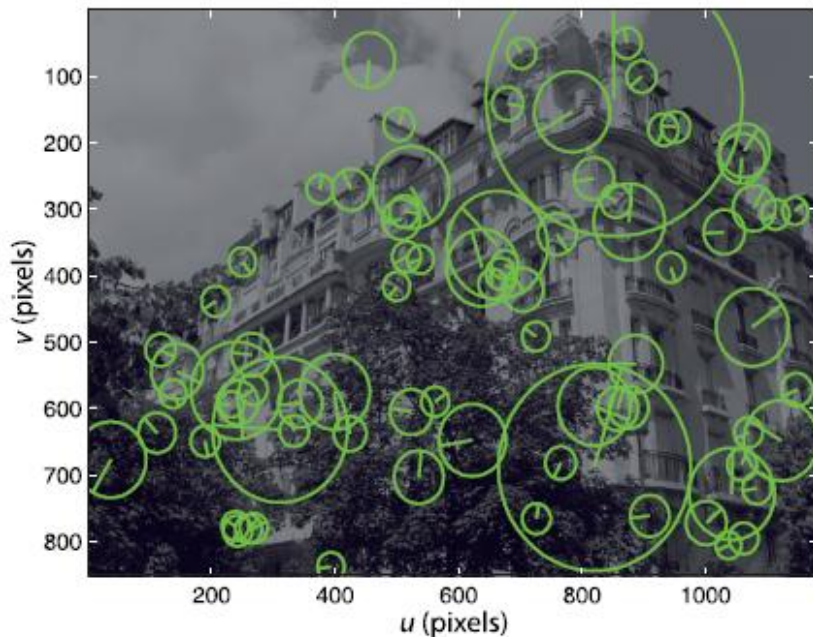
Mesma imagem com cantos detectados pelo detector de Harris



Detectores de Cantos no Espaço de Escalas (Scale-Space)

SURF – Speeded-Up Robust Feature

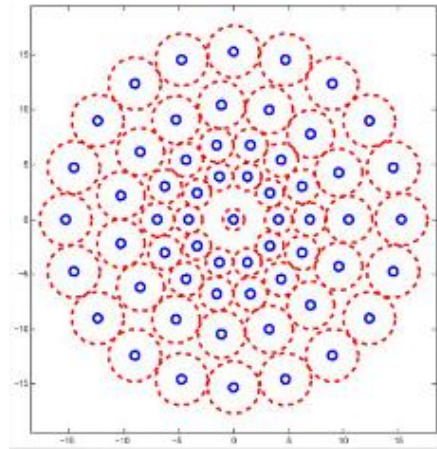
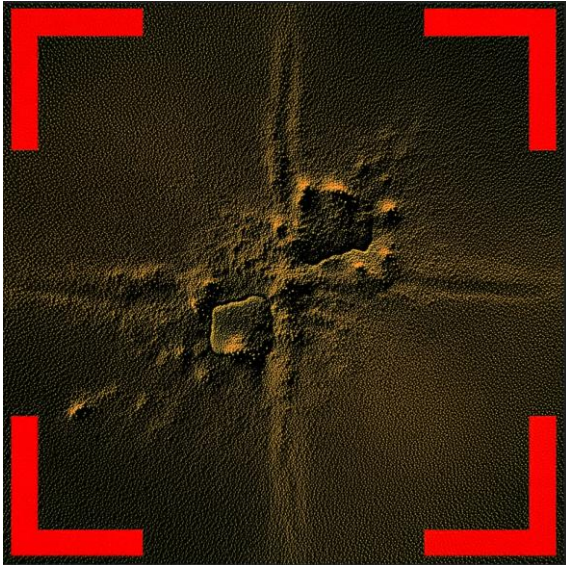
- Pode ser visto como um *descriptor* de gradiente em sub-regiões.
- **Robusto a variações de brilho, escala e rotação**



Histograma de número de pontos característicos pela escala em escala logarítmica

Detectores de Cantos no Espaço de Escalas (Scale-Space)

BRISK – (Binary Robust Invariant Scalable Keypoints)



- **Deteção de Cantos:** BRISK usa o detector AGAST (Adaptive and Generic Accelerated Segment Test) para detecção rápida de cantos.
- **Descritor:** Utiliza um descritor binário, que é muito mais rápido de calcular e comparar do que os descritores SIFT e SURF.
- **Vantagens:** Muito rápido e eficiente em termos de memória, adequado para aplicações em tempo real.
- **Desvantagens:** Menos preciso que SIFT e SURF em algumas situações.

Detectores de Cantos no Espaço de Escalas (Scale-Space)

Comparativo

- **SIFT** é o mais preciso e robusto, mas também o mais lento.
- **SURF** oferece um bom equilíbrio entre velocidade e precisão.
- **BRISK** é o mais rápido, ideal para aplicações em tempo real, mas pode ser menos preciso.

Característica	SIFT	SURF	BRISK
Detecção de pontos-chave	DoG	Determinante Hessiano	AGAST
Descritor	Gradientes orientados	Haar wavelets	Binário
Velocidade	Lento	Médio	Rápido
Precisão	Alta	Média	Média
Robustez	Alta	Média	Média
Patenteado	Sim	Não	Não

- Há outros detectores de cantos, como **FAST**, **BRIEF**, **FREAK**, **ORB (Oriented Fast + Rotated BRIEF)** etc.

EXTRAÇÃO DE TIPOS SUPERFÍCIE EM IMAGENS DE PROFUNDIDADE

- Objetos 3-D podem ser descritos em termos da forma e posição das superfícies de que são feitos.
- Por exemplo: Cone \implies 1 superfície cônica
1 superfície plana
- Descrições baseadas em superfície podem ser usadas para:
 - Classificação
 - Estimativa de Posição
 - Engenharia Reversa
 - Uso generalizado em Computação Gráfica
- Imagens de profundidade são a versão digitalizada (amostrada) das superfícies visíveis na cena
 - A forma da superfície na imagem e a forma das superfícies na cena visível são as mesmas (desprezando distorções).

SEGMENTAÇÃO E CLASSIFICAÇÃO DE SUPERFÍCIES

- Uma das formas de classificação de superfícies é a **Segmentação H-K**, que utiliza as *curvaturas média (H)* e *gaussiana (K)* para classificação de superfícies, especialmente em imagens de profundidade (x,y,z).
- A técnica se baseia na **geometria diferencial** para analisar a forma local de uma superfície em cada ponto.
- As **curvaturas H e K** são calculadas a partir das derivadas da superfície e fornecem informações importantes sobre a sua curvatura. Envolvem princípios fundamentais em muitos algoritmos em VC.
- Deep Learning é outra abordagem mais moderna para segmentação.

Segmentação H-K de Imagens de Profundidade

Dada uma imagem I na forma r_{ij} , compute uma nova imagem registrada com I (mesmo sistema de coordenadas), do mesmo tamanho, em que cada pixel é associado com uma classe de forma local selecionada de um dado dicionário

São necessárias as seguintes ferramentas:

- Dicionário de classes de formas
- Algoritmo determinando que classe de forma melhor aproxima da superfície em cada pixel
- Algoritmos para cálculo de vetores normais e curvaturas de superfícies a partir de nuvem de pontos

- Definindo Classes de Formas (Perfis)
 - Necessária definição local da forma
 - Teoria vem da Geometria Diferencial
 - Usam-se os sinais da curvatura H média e da curvatura K Gaussiana

- Segmentação H-K

- Particiona uma imagem de profundidade em regiões de forma homogênea, chamadas “pedaços (ou segmentos) de superfícies” (surface patches). Base em geometria diferencial.

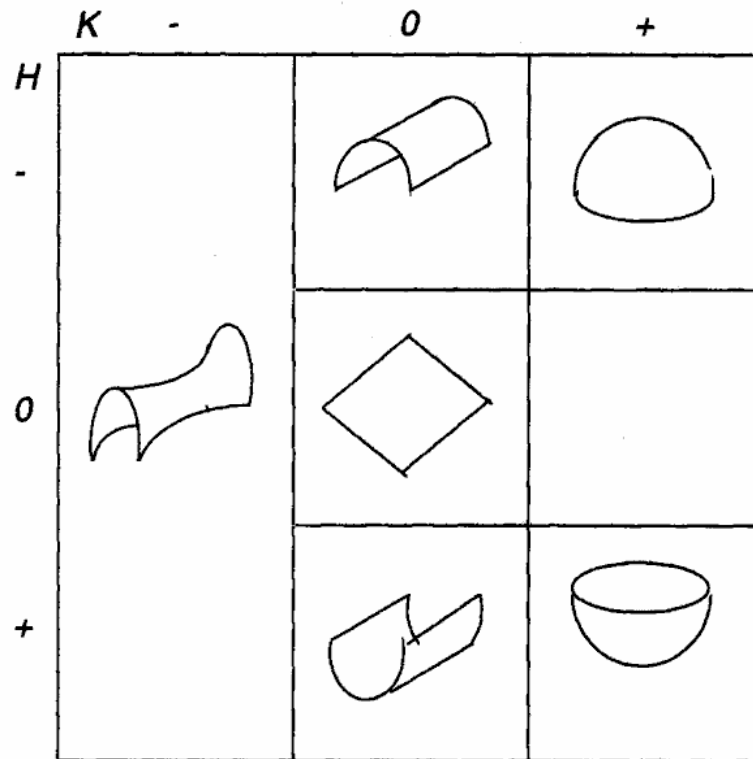
H = curvatura média;

K = curvatura gaussiana

$$K = k_1 \cdot k_2$$

$$H = \frac{k_1 + k_2}{2}$$

k_1, k_2 = máxima e mínima curvaturas da superfície em um ponto



Côncava: $k > 0$

Convexa: $k < 0$

Ilustração das formas locais resultantes da classificação H-K.

- Geometria Diferencial Elementar

- A parametrização de uma superfície mapeia pontos (u,v) no domínio de pontos p no espaço.

$$p(u, v) = (x(u, v), y(u, v), z(u, v))$$

e $\left. \begin{matrix} p_u(u, v) \\ p_v(u, v) \end{matrix} \right\} \rightarrow$ primeiras derivadas são vetores ortogonais que varrem o plano tangente à superfície no ponto $(x, y, z) = p(u, v)$

- A normal à superfície no ponto p é

$$\vec{n}(p) = \frac{\vec{p}_u \times \vec{p}_v}{\|\vec{p}_u \times \vec{p}_v\|}$$

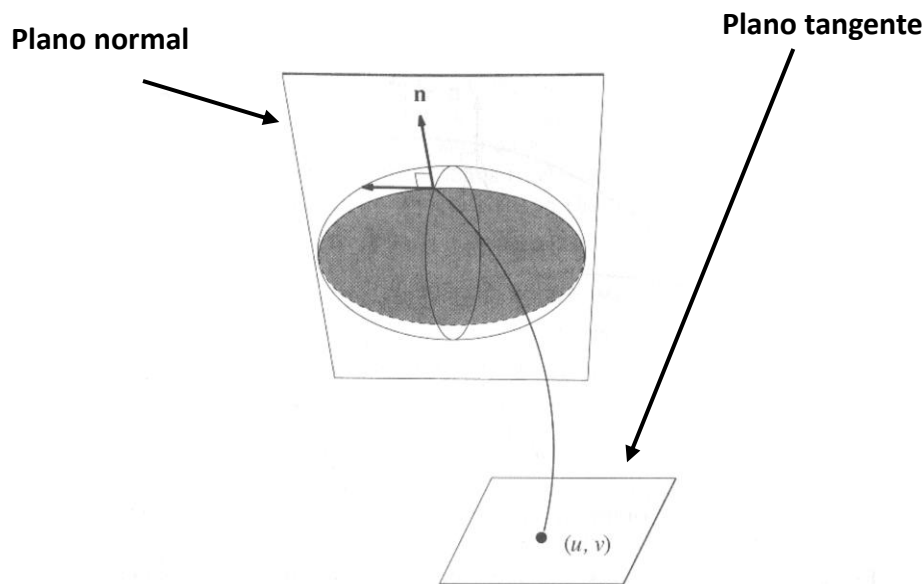
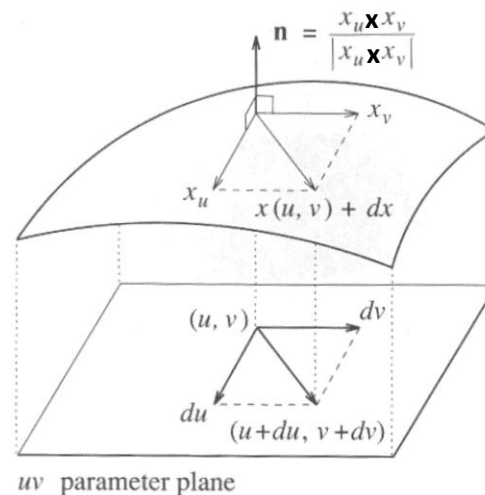


Diagrama ilustrando a geometria da curva que passa por um ponto em um plano interceptante (plano normal) que contém o vetor normal à superfície no ponto.



Uma superfície definida parametricamente que mostra o plano tangente e a normal à superfície.

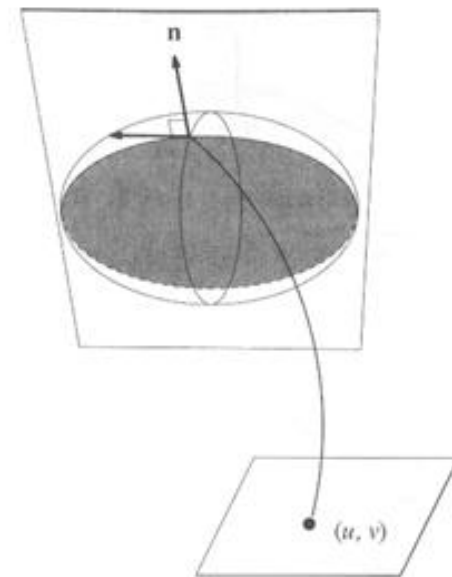
- Um plano normal à superfície passando por \mathbf{p} forma uma curva plana e contém \mathbf{n} ;
- Passando o plano normal por \mathbf{p} em todas as direções têm-se diferentes curvas planas;
- k_1 é a máxima curvatura de uma curva plana e k_2 a mínima
- Define-se

$$K = k_1 \cdot k_2$$

$$H = \frac{k_1 + k_2}{2}$$

em que $k = \frac{y''}{(1 + (y')^2)^{3/2}} = (1/\text{raio de curvatura})$

(se a superfície é convexa $k < 0$, se côncava $k > 0$)



- Estimativa de Formas Locais

- Para uma imagem de profundidade r_{ij} , $(x,y,h(x,y))$

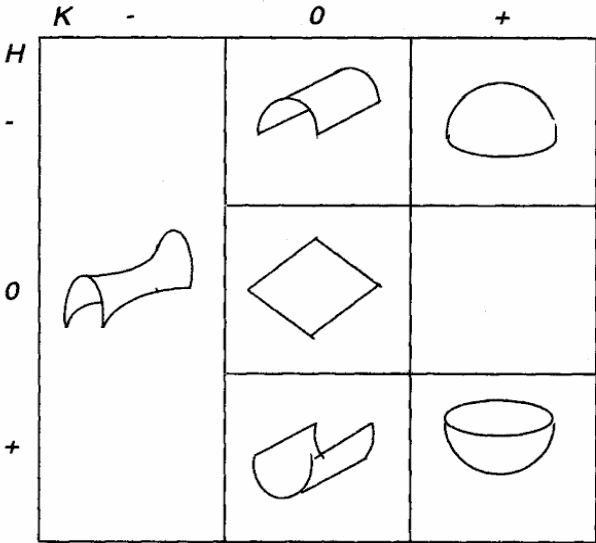


Ilustração de formas locais resultantes da classificação H-K

<i>K</i>	<i>H</i>	Local shape class
0	0	plane
0	+	concave cylindrical
0	−	convex cylindrical
+	+	concave elliptic
+	−	convex elliptic
−	any	hyperbolic

Esquema de classificação de entalhes

$$K = k_1.k_2$$

$$H = \frac{k_1 + k_2}{2}$$

- Estimativas de Formas Locais

- Para uma imagem de profundidade $r_{i,j}$, $(x, y, h(x, y))$,

$$K = \frac{h_{xx} \cdot h_{yy} - h_{xy}^2}{(1 + h_x^2 + h_y^2)^2}$$

$$2.H = \frac{(1 + h_x^2) \cdot h_{yy} - 2 \cdot h_x \cdot h_y \cdot h_{xy} + (1 + h_y^2) \cdot h_{xx}}{(1 + h_x^2 + h_y^2)^{3/2}}$$

onde os índices indicam derivações parciais.

Entretanto, alguns pontos têm de ser considerados:

- 1) **As imagens contêm ruídos** que precisam ser atenuados, para reduzir as distorções numéricas nos cálculos das derivadas e curvaturas (o pior ruído é o da quantização ou da precisão limitada do sensor de profundidade)
- 2) Os resultados ainda podem conter **pequenos defeitos** em função do ruído, e que podem ser eliminados por filtragem adicional (morfologia de imagens ==> shrinking and expansion)
- 3) Retalhos planos deveriam ter $H = K = 0$, mas estimativas numéricas de H e K **nunca são exatamente zero**. Portanto limiares-zero para H e K devem ser estabelecidos. A precisão de extração de retalhos planos depende do ruído, orientação do plano e limiares $p/H-K$.
- 4) Estimativas de derivadas e curvaturas não fazem sentido em **descontinuidades**. Para evitar descontinuidades, deve-se utilizar um detector de contornos de degraus (CANNY_EDGE_DETECTOR), identificar as bordas e retirá-las (descontinuidades) da superfície.

Segmentação H-K

VANTAGENS

- **Invariância a rotação e translação:** As curvaturas H e K são invariantes a rotações e translações da superfície, tornando a técnica robusta a diferentes posturas do objeto.
- **Descrição da forma local:** Fornecem uma descrição precisa da forma da superfície em cada ponto.

DESVANTAGENS

- **Sensibilidade a ruído:** O cálculo de H e K pode ser sensível a ruído nos dados de profundidade.
- **Complexidade computacional:** A estimação precisa das curvaturas pode ser computacionalmente cara.

APLICAÇÕES

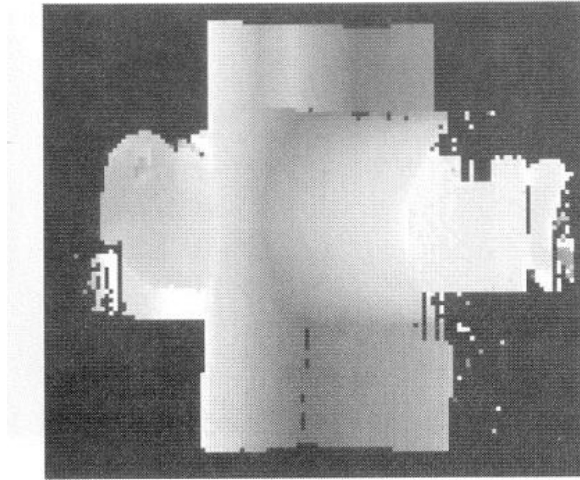
- **Reconhecimento de objetos:** Classificar objetos 3D com base em suas formas.
- **Segmentação de imagens:** Dividir a imagem em regiões com diferentes tipos de superfície.
- **Reconstrução 3D:** Melhorar a qualidade de modelos 3D reconstruídos a partir de imagens de profundidade.

Algoritmos RANGE_SURF_PATCHES

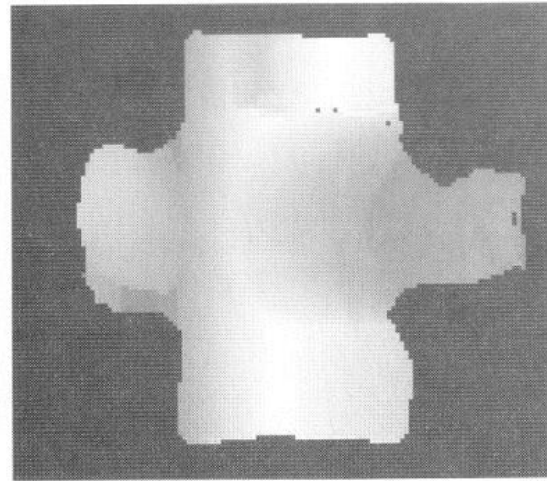
Entrada é I (imagem de superfícies), na **forma r_{ij}** , e um conjunto de rótulos de formas $[s_1, \dots, s_6]$ associados às classes da lista.

- 1) Aplicar atenuação Gaussiana a I , obtendo-se I_s .
- 2) Calcular as derivadas das imagens $I_x, I_y, I_{xy}, I_{xx}, I_{yy}$.
- 3) Calcular H e K (p/ imagem de profund.).
- 4) Calcular a forma da imagem, S , atribuindo um rótulo de imagem, S_i , a cada pixel, de acordo com as regras da lista.

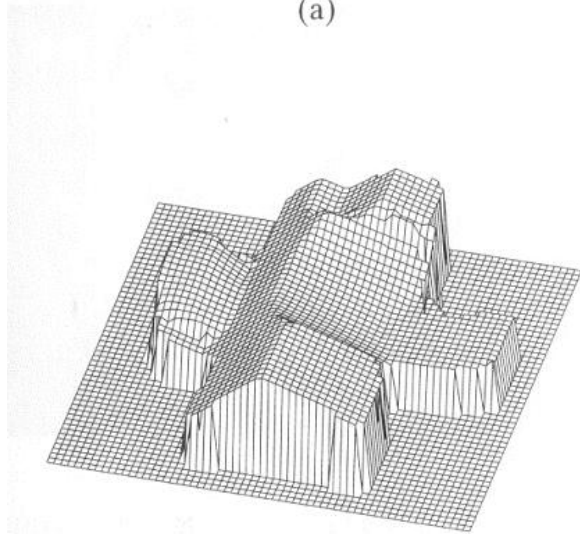
A saída é a forma de imagem S .



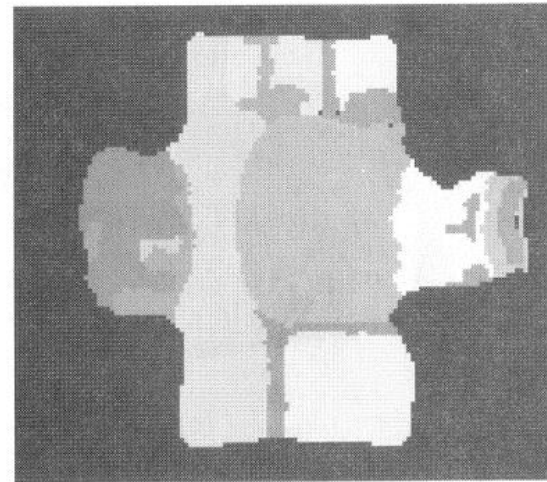
(a)



(b)

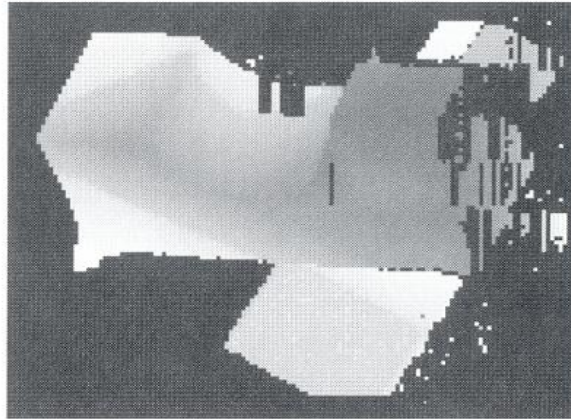


(c)

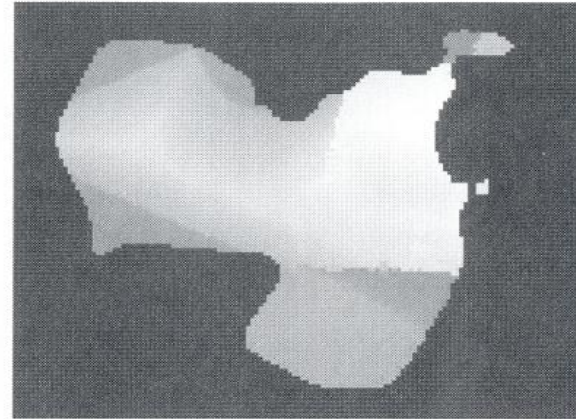


(d)

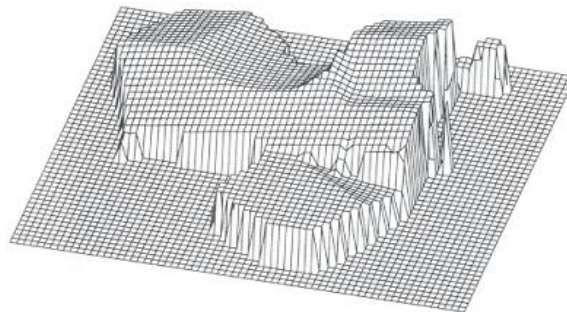
(a): Imagem de profundidade, em tom de cinza (mais escuro mais próximo ao sensor). (b): depois de suavização, em tom de cinza. (c): o mesmo que *b*, como plotagem isométrica 3-D. (d): Segmentos detectados pela segmentação H-K



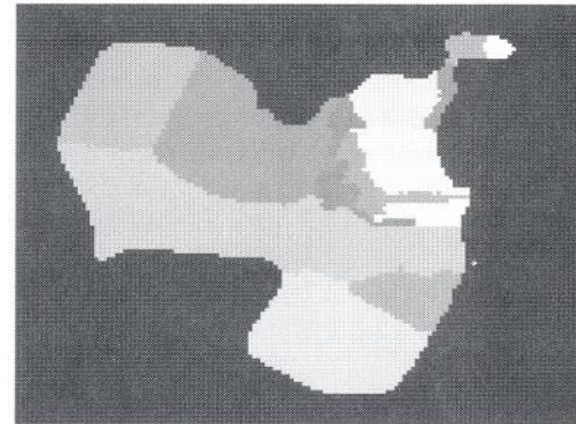
(a)



(b)



(c)



(d)

(a): Imagem de profundidade, em tom de cinza (mais escuro mais próximo ao sensor). (b): depois de suavização, em tom de cinza. (c): o mesmo que *b*, como plotagem isométrica 3-D. (d): Segmentos detectados pela segmentação H-K