

Detecção Robusta de Movimento de Câmera em Vídeos por Análise de Fluxo Ótico Ponderado

Rodrigo Minetto

Prof. Dr. Neucimar Jerônimo Leite (Orientador)

Prof. Dr. Jorge Stolfi (Co-orientador)

Instituto de Computação / Universidade Estadual de Campinas (UNICAMP)



Campinas, 17 de Agosto de 2007

Sumário da Apresentação

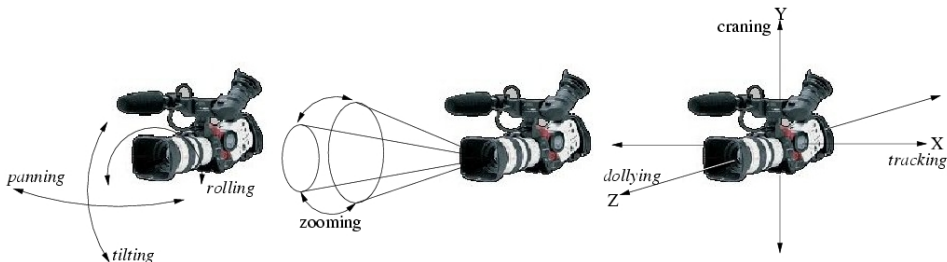
- Introdução;
- Conceitos;
- Algoritmo Proposto;
- Resultados;
- Conclusões e Trabalhos Futuros.

Introdução

- 1 Contexto e motivação;
- 2 Nossa contribuição.

Contexto e motivação

- O grande volume de vídeos digitais requer métodos eficientes para navegação, busca e visualização do seu conteúdo;
- Para recuperar informações relevantes, o passo fundamental consiste da divisão do vídeo em segmentos coerentes;
- Os movimentos de câmera são relevantes para essa tarefa, pois revelam algum foco de interesse do operador.



Contexto e motivação

- Em filmagens, tais como videoconferências, os movimentos de câmera são a principal característica para segmentação;



- A segmentação permite a organização da gravação em diversos cliques de vídeo, um para cada orador;
- A grande dificuldade neste problema é distinguir movimentos da câmera do movimento de objetos e pessoas na cena.

Nossa contribuição

Como veremos a seguir:

- Um algoritmo original para detectar automaticamente movimentos da câmera (tilt/pan/roll/zoom) em vídeos digitais;
- Aplicável a vídeos criados por uma câmera em posição fixa que pode ser apontada em várias direções através de movimentos de rotação e mudanças de ampliação da lente;
- O algoritmo proposto provê uma análise precisa e confiável da quantidade de movimento existente entre dois quadros consecutivos quaisquer;
- Este consegue em muitos casos distinguir movimentos de câmera de movimentos de objetos na cena.

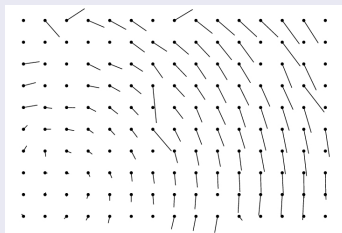
Conceitos

- 1 Fluxo ótico;
 - 1 Conceito;
 - 2 Problemas da definição do fluxo ótico;
 - 3 Algoritmos para o cálculo de fluxo ótico;
- 2 Fluxos de movimentos de câmera;
 - 1 Modelos;
 - 2 Fluxos canônicos.

Fluxo Ótico

Conceito

- A maioria das técnicas existentes para determinar movimentos de câmera baseia-se na análise do fluxo ótico;
- O fluxo ótico é um função que associa a cada ponto p de uma imagem seu deslocamento ótico $f(p)$;

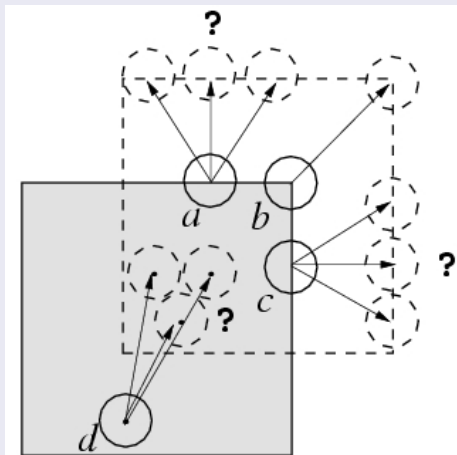


- O correspondente ótico de p é um ponto q cujos pixels em uma vizinhança são maximalmente similares aos valores de p ;
- O deslocamento ótico é o vetor $f = p - q$.

Fluxo Ótico

Problemas da definição de fluxo ótico

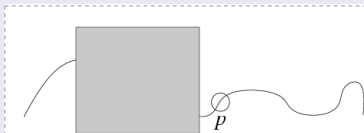
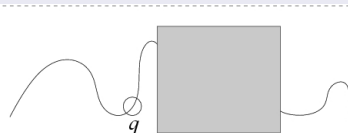
- Ambigüidades na correspondência ótica:



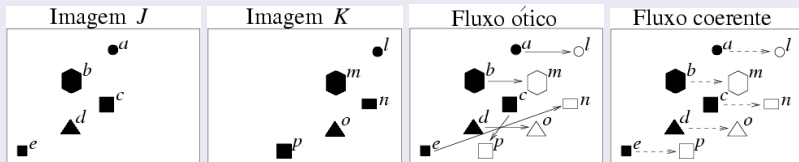
Fluxo Ótico

Problemas da definição de fluxo ótico

- Problema da oclusão:

Imagem: J Imagem: K

- Problema da incoerência espacial:



Fluxo Ótico

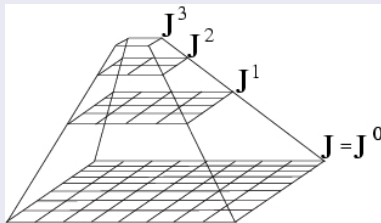
Algoritmos para o cálculo do fluxo ótico

- Algoritmo de Busca exaustiva;
- Algoritmo de Horn e Schunck (HS);
- Algoritmo de Lucas e Kanade (LK);
- Algoritmo de Kanade, Lucas e Tomasi (KLT);
- Fluxo ótico do MPEG.

Fluxo Ótico

Algoritmo de Kanade, Lucas e Tomasi (KLT)

- Resolve o problema de rastreamento em múltiplas escalas de resolução, começando com uma versão de baixa resolução da imagem dada e passando sucessivamente a versões mais detalhadas;



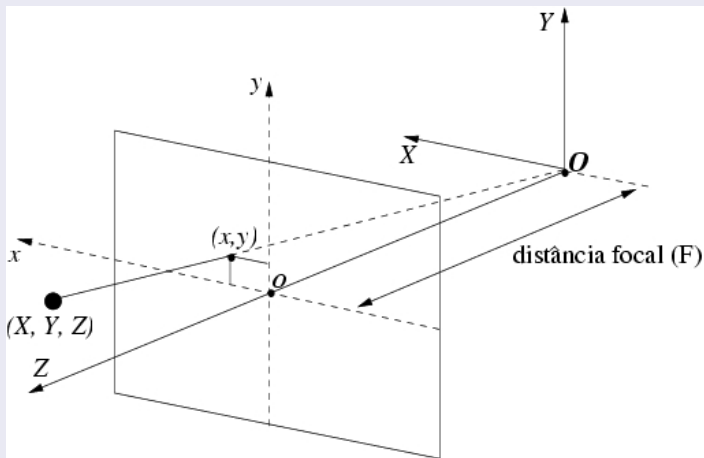
- Utiliza o algoritmo LK em cada resolução;
- Permite calcular grandes deslocamentos.

Fluxo de movimentos de câmara

- A detecção automática de movimentos da câmara baseia-se no princípio de que estes movimentos geralmente produzem fluxos óticos característicos nas imagens;
- Fluxos estes bem distintos dos fluxos resultantes de movimentos de objetos na cena;

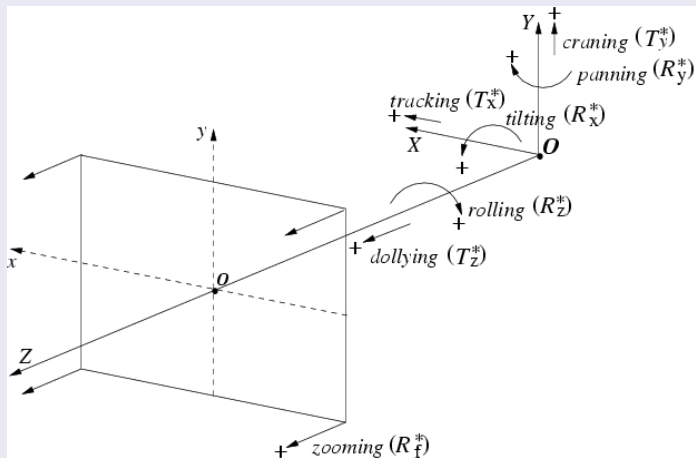
Fluxos de movimentos de câmera

O sistema canônico para projeção perspectiva plana



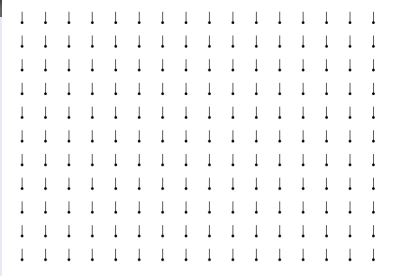
Fluxos de movimentos de câmera

Velocidades que definem o movimento instantâneo da câmera



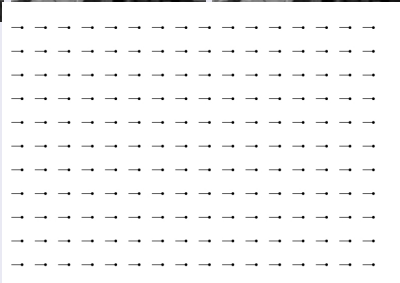
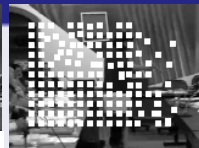
Fluxos de movimentos de câmera

Fluxos canônicos devido a tilt $\{r(x, y) = (0, 1)\}$



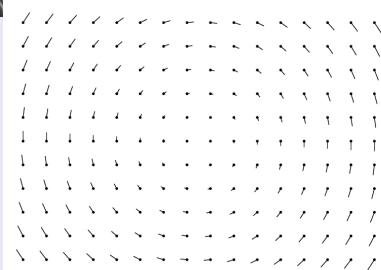
Fluxos de movimentos de câmera

Fluxos canônico devido a pan $\{r(x, y) = (-1, 0)\}$



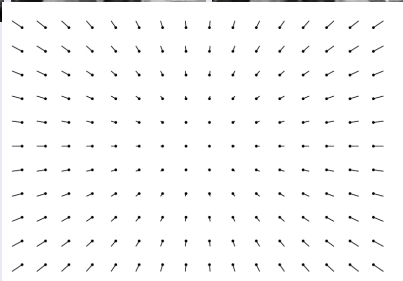
Fluxos de movimentos de câmera

Fluxo canônico devido a roll $\{r = (v, x)\}$



Fluxos de movimentos de câmera

Fluxo canônico devido a zoom $\{r_t = (x, y)\}$



Algoritmo Proposto

- 1 Características;
- 2 Pseudo-Código.

Características

WOFF

- 1 O algoritmo objetiva extrair do fluxo ótico os parâmetros de movimento da câmera R_x , R_y , R_z e R_f (tilt, pan, roll e zoom), supondo que a câmera é estacionária ($T_x = T_y = T_z = 0$).
- 2 a entrada é um fluxo ótico amostrado, cujos vetores são ponderados de acordo com sua confiabilidade;
- 3 o algoritmo usa um procedimento de mínimos quadrados ponderado para obter o fluxo ideal que melhor se ajusta ao fluxo observado.
- 4 os pesos dos vetores do fluxo são ajustados iterativamente com base na análise estatística do fluxo residual, para distinguir entre movimentos da câmera e movimentos de objetos na cena.

WOFF(p, f, κ, n)

Dados: uma lista p de n pontos, um fluxo ótico f amostrado em p , e uma lista κ de escores.

Resultado: parâmetros R_x, R_y, R_z, R_f e pesos finais w_1, \dots, w_n para os pontos.

Ajuste inicial de pesos:

1. **para** $i \in \{0, 1, \dots, n-1\}$ **faça** $w_i \leftarrow \kappa_i / \sqrt{|f_i|^2 + \epsilon^2}$;
2. **repita** {
3. convergiu \leftarrow **verdadeiro**;
4. $R_x, R_y, R_z, R_f \leftarrow$ **AnáliseMínimosQuadrados** (f, p, w);
 Cálculo dos resíduos:
5. **para** $i \in \{0, 1, \dots, n-1\}$ **faça** $d_i \leftarrow f_i - \tilde{f}(R_x, R_y, R_z, R_f)(p_i)$;
6. $\mu \leftarrow$ Média (d, w); $\sigma \leftarrow$ Desvio (d, w, μ);
 Ajuste iterativo:
7. **para** $i \in \{0, 1, \dots, n-1\}$ **faça** {
8. **se** ($w_i \neq 0$) e ($|d_i| > 3\sigma$) **então** {
9. $w_i \leftarrow 0$;
10. convergiu \leftarrow **falso**;
11. }
12. }
13. **enquanto** (**não** convergiu);
14. **retorne** R_x, R_y, R_z, R_f, w ;

Testes da detecção de movimento

- 1 Vídeos de teste;
- 2 Experimentos;
- 3 Limitações;
- 4 Performance.

Vídeos de teste

- Para avaliar a performance do método proposto, todos os vídeos utilizados foram segmentados manualmente na íntegra;
- Nove vídeos reais foram utilizados (3 do CONSU e 6 da internet);
- Todos os pares de quadros consecutivos em cada vídeo foram classificados manualmente nas duas classes, pares com *movimento de câmera* (M) e pares com *câmera estacionária* (S);
- O fluxo foi amostrado em uma grade regular de 16×11 pontos;
- Os vídeos foram reduzidos em escala;
- O tempo de execução em um processador Intel pentium 4 de 3.0GHz com 2GB de memória para um fluxo de 16×11 pontos foi de 60 milissegundos em média.

Vídeos de teste

- Para avaliar a performance do método proposto, todos os vídeos utilizados foram segmentados manualmente na íntegra;
- Nove vídeos reais foram utilizados (3 do CONSU e 6 da internet);
- Todos os pares de quadros consecutivos em cada vídeo foram classificados manualmente nas duas classes, pares com *movimento de câmera* (M) e pares com *câmera estacionária* (S);
- O fluxo foi amostrado em uma grade regular de 16×11 pontos;
- Os vídeos foram reduzidos em escala;
- O tempo de execução em um processador Intel pentium 4 de 3.0GHz com 2GB de memória para um fluxo de 16×11 pontos foi de 60 milissegundos em média.

Vídeos de teste

- Para avaliar a performance do método proposto, todos os vídeos utilizados foram segmentados manualmente na íntegra;
- Nove vídeos reais foram utilizados (3 do CONSU e 6 da internet);
- Todos os pares de quadros consecutivos em cada vídeo foram classificados manualmente nas duas classes, pares com *movimento de câmera* (M) e pares com *câmera estacionária* (S);
- O fluxo foi amostrado em uma grade regular de 16×11 pontos;
- Os vídeos foram reduzidos em escala;
- O tempo de execução em um processador Intel pentium 4 de 3.0GHz com 2GB de memória para um fluxo de 16×11 pontos foi de 60 milissegundos em média.

Vídeos de teste

- Para avaliar a performance do método proposto, todos os vídeos utilizados foram segmentados manualmente na íntegra;
- Nove vídeos reais foram utilizados (3 do CONSU e 6 da internet);
- Todos os pares de quadros consecutivos em cada vídeo foram classificados manualmente nas duas classes, pares com *movimento de câmera* (M) e pares com *câmera estacionária* (S);
- O fluxo foi amostrado em uma grade regular de 16×11 pontos;
- Os vídeos foram reduzidos em escala;
- O tempo de execução em um processador Intel pentium 4 de 3.0GHz com 2GB de memória para um fluxo de 16×11 pontos foi de 60 milissegundos em média.

Vídeos de teste

- Para avaliar a performance do método proposto, todos os vídeos utilizados foram segmentados manualmente na íntegra;
- Nove vídeos reais foram utilizados (3 do CONSU e 6 da internet);
- Todos os pares de quadros consecutivos em cada vídeo foram classificados manualmente nas duas classes, pares com *movimento de câmera* (M) e pares com *câmera estacionária* (S);
- O fluxo foi amostrado em uma grade regular de 16×11 pontos;
- Os vídeos foram reduzidos em escala;
- O tempo de execução em um processador Intel pentium 4 de 3.0GHz com 2GB de memória para um fluxo de 16×11 pontos foi de 60 milissegundos em média.

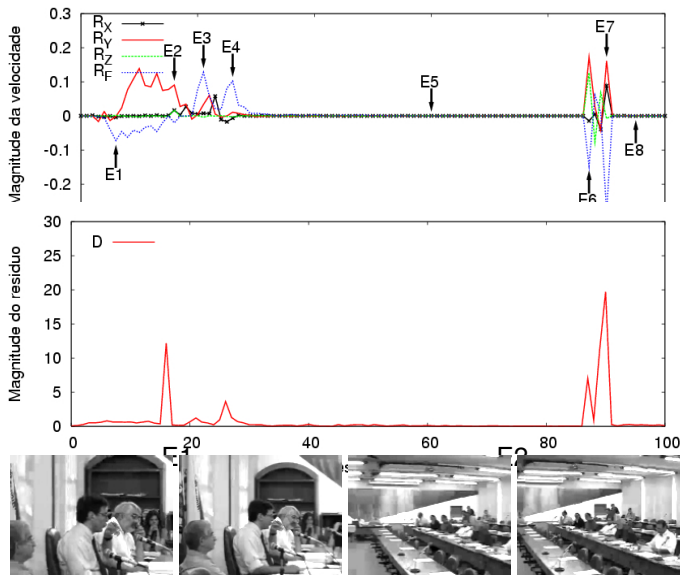
Vídeos de teste

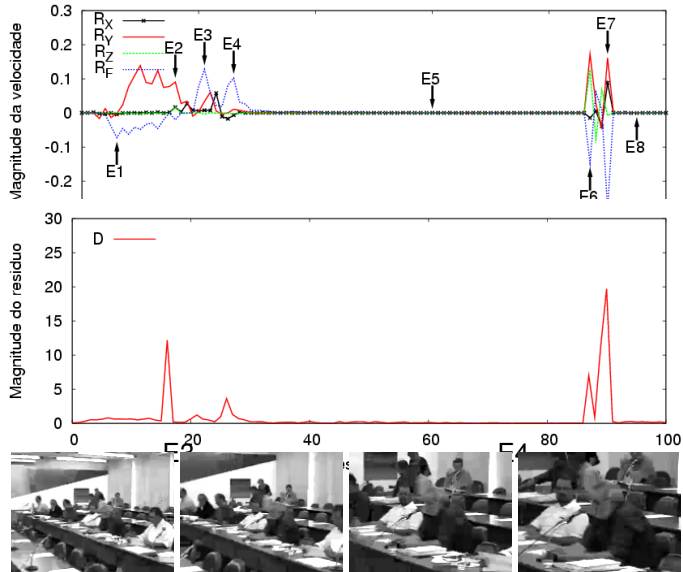
- Para avaliar a performance do método proposto, todos os vídeos utilizados foram segmentados manualmente na íntegra;
- Nove vídeos reais foram utilizados (3 do CONSU e 6 da internet);
- Todos os pares de quadros consecutivos em cada vídeo foram classificados manualmente nas duas classes, pares com *movimento de câmera* (M) e pares com *câmera estacionária* (S);
- O fluxo foi amostrado em uma grade regular de 16×11 pontos;
- Os vídeos foram reduzidos em escala;
- O tempo de execução em um processador Intel pentium 4 de 3.0GHz com 2GB de memória para um fluxo de 16×11 pontos foi de 60 milissegundos em média.

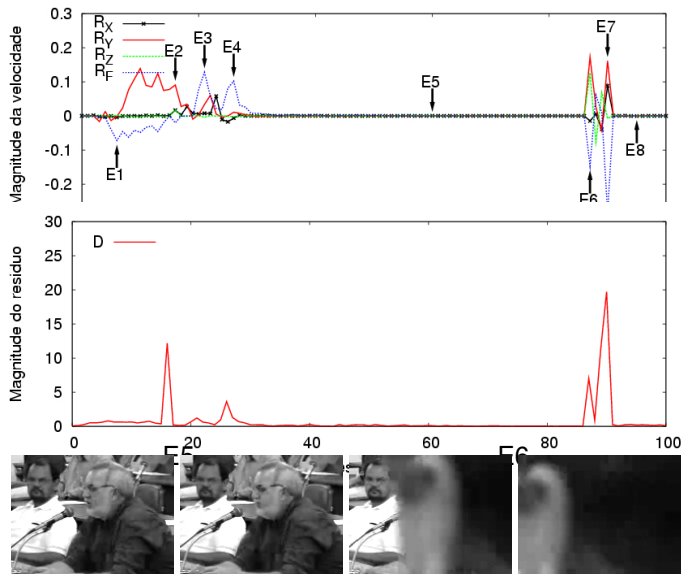
Vídeos utilizados nos experimentos

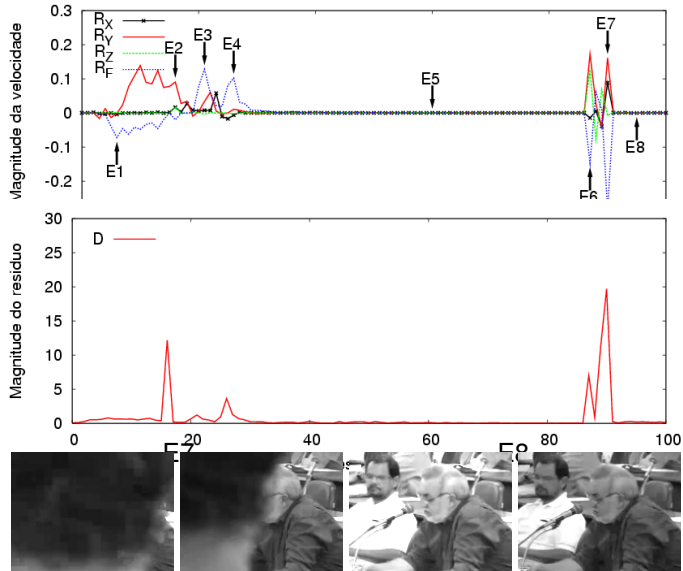
Vídeos utilizados

Vídeo	Tamanho		Tipo	Dimensões (quadro)	Trans. de Cenas	Pares M
	(minutos)	(quadros)				
consu1	00:39:27	59140	RP	160 × 120	80	1360
consu2	00:11:04	16600	RP	160 × 120	23	688
consu3	01:38:38	147959	RP	160 × 120	192	4459
montanha1	00:00:38	341	AVI	160 × 108	3	119
península	00:00:17	195	AVI	160 × 108	2	50
montanha2	00:00:16	183	AVI	160 × 108	2	45
cidade	00:00:19	247	AVI	160 × 108	2	109
feira	00:00:21	261	AVI	160 × 108	4	122
fazenda	00:00:14	177	AVI	160 × 108	1	53

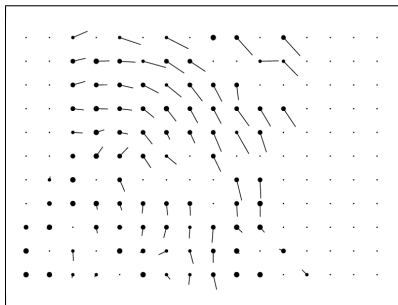




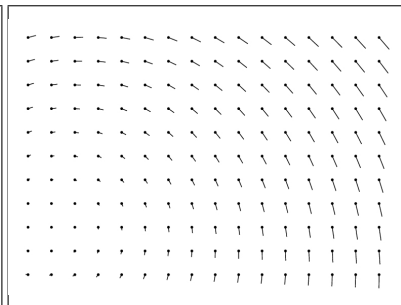




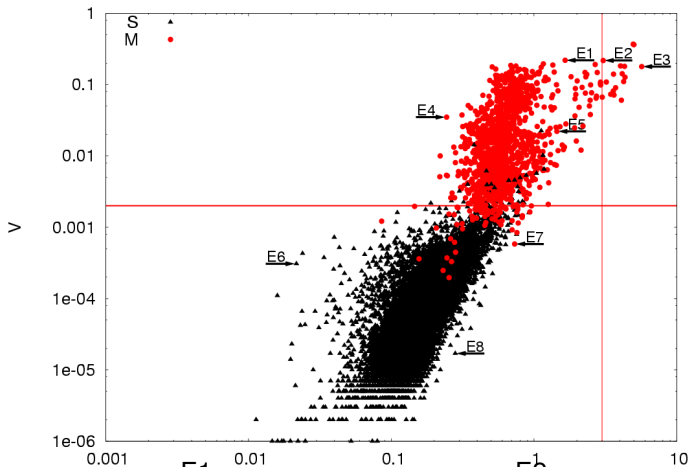
Limitação (Erro de classificação)

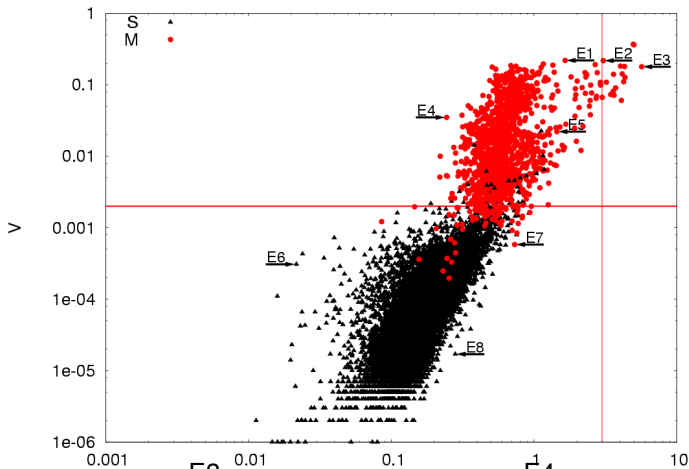


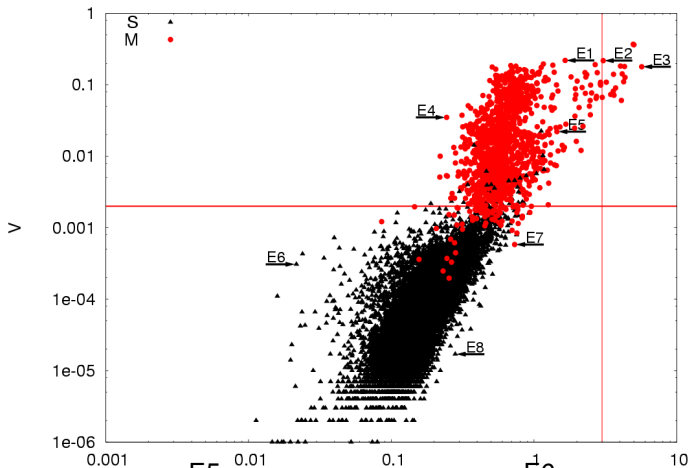
f (pesos finais)

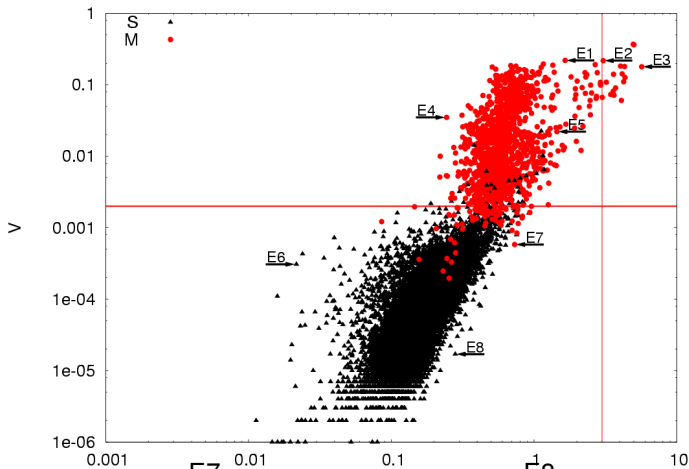


\tilde{f} (fluxo estimado)









Desempenho do detector de movimento de câmera

O desempenho foi quantificado através das métricas:

$$pr = \frac{V_+}{V_+ + F_+} \text{ (Precisão)} \quad rc = \frac{V_+}{V_+ + F_-} \text{ (Revocação)}$$

- V_+ : positivos verdadeiros (pares de quadro com câmera em movimento corretamente identificados);
- F_+ : positivos falsos (pares com câmera estacionária identificados como câmera em movimento);
- F_- : negativos falsos (pares com câmera em movimento identificados como câmera estacionária).

Tabela de performance (WOFF)

Vídeo	V_+	F_+	F_-	pr	rc
consu1	1263	97	45	0.97	0.93
consu2	637	51	8	0.99	0.93
consu3	4182	277	450	0.90	0.94
montanha1	115	4	0	1.00	0.97
península	49	1	0	1.00	0.98
montanha2	45	0	0	1.00	1.00
cidade	109	0	0	1.00	1.00
feira	122	0	0	1.00	1.00
fazenda	53	0	0	1.00	1.00

Performance para cinco variantes do algoritmo WOFF

Vídeo	WOFF		Teste1		Teste2		Teste3		Teste4	
	<i>pr</i>	<i>rc</i>	<i>pr</i>	<i>rc</i>	<i>pr</i>	<i>rc</i>	<i>pr</i>	<i>rc</i>	<i>pr</i>	<i>rc</i>
consu1	0.97	0.93	0.22	0.94	0.90	0.94	0.58	0.97	0.97	0.83
consu2	0.99	0.93	0.28	0.95	0.98	0.94	0.67	0.98	0.99	0.88
consu3	0.90	0.94	0.30	0.90	0.86	0.93	0.42	0.96	0.94	0.85

- Teste1: Avaliando a importância dos pesos;
- Teste2: Avaliando a importância do ajuste iterativo;
- Teste3: Avaliando a importância do escore de confiabilidade;
- Teste4: Avaliando a importância do ajuste inicial de pesos.

Conclusões e Trabalhos Futuros

Conclusões e Trabalhos Futuros

- Proposta de um algoritmo original para detectar movimentos de câmera;
- Este algoritmo realiza uma análise robusta da quantidade de movimento existente entre dois quadros de um vídeo;
- Testes com vídeos reais resultaram em taxas de acerto acima de 90%;
- Um estudo sobre importantes algoritmos para o cálculo de fluxo ótico e sobre fluxos canônicos devido a movimento de câmera;
- Um projeto interessante é estender o algoritmo WOFF para situações com câmera em movimento (T_x , T_y ou T_z diferente de zero);
- Utilizar a informação de fluxo direcional na composição de imagens panorâmicas.

Conclusões e Trabalhos Futuros

- Proposta de um algoritmo original para detectar movimentos de câmera;
- Este algoritmo realiza uma análise robusta da quantidade de movimento existente entre dois quadros de um vídeo;
- Testes com vídeos reais resultaram em taxas de acerto acima de 90%;
- Um estudo sobre importantes algoritmos para o cálculo de fluxo ótico e sobre fluxos canônicos devido a movimento de câmera;
- Um projeto interessante é estender o algoritmo WOFF para situações com câmera em movimento (T_x , T_y ou T_z diferente de zero);
- Utilizar a informação de fluxo direcional na composição de imagens panorâmicas.

Conclusões e Trabalhos Futuros

- Proposta de um algoritmo original para detectar movimentos de câmera;
- Este algoritmo realiza uma análise robusta da quantidade de movimento existente entre dois quadros de um vídeo;
- Testes com vídeos reais resultaram em taxas de acerto acima de 90%;
- Um estudo sobre importantes algoritmos para o cálculo de fluxo ótico e sobre fluxos canônicos devido a movimento de câmera;
- Um projeto interessante é estender o algoritmo WOFF para situações com câmera em movimento (T_x , T_y ou T_z diferente de zero);
- Utilizar a informação de fluxo direcional na composição de imagens panorâmicas.

Conclusões e Trabalhos Futuros

- Proposta de um algoritmo original para detectar movimentos de câmera;
- Este algoritmo realiza uma análise robusta da quantidade de movimento existente entre dois quadros de um vídeo;
- Testes com vídeos reais resultaram em taxas de acerto acima de 90%;
- Um estudo sobre importantes algoritmos para o cálculo de fluxo ótico e sobre fluxos canônicos devido a movimento de câmera;
- Um projeto interessante é estender o algoritmo WOFF para situações com câmera em movimento (T_x , T_y ou T_z diferente de zero);
- Utilizar a informação de fluxo direcional na composição de imagens panorâmicas.

Conclusões e Trabalhos Futuros

- Proposta de um algoritmo original para detectar movimentos de câmera;
- Este algoritmo realiza uma análise robusta da quantidade de movimento existente entre dois quadros de um vídeo;
- Testes com vídeos reais resultaram em taxas de acerto acima de 90%;
- Um estudo sobre importantes algoritmos para o cálculo de fluxo ótico e sobre fluxos canônicos devido a movimento de câmera;
- Um projeto interessante é estender o algoritmo WOFF para situações com câmera em movimento (T_x , T_y ou T_z diferente de zero);
- Utilizar a informação de fluxo direcional na composição de imagens panorâmicas.

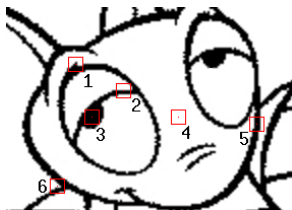
Conclusões e Trabalhos Futuros

- Proposta de um algoritmo original para detectar movimentos de câmera;
- Este algoritmo realiza uma análise robusta da quantidade de movimento existente entre dois quadros de um vídeo;
- Testes com vídeos reais resultaram em taxas de acerto acima de 90%;
- Um estudo sobre importantes algoritmos para o cálculo de fluxo ótico e sobre fluxos canônicos devido a movimento de câmera;
- Um projeto interessante é estender o algoritmo WOFF para situações com câmera em movimento (T_x , T_y ou T_z diferente de zero);
- Utilizar a informação de fluxo direcional na composição de imagens panorâmicas.

Dúvidas?

Escore de confiabilidade

$$\kappa_j = \frac{\min(\lambda'(p_i), \lambda''(p_i))}{\tau(p_i)}$$


 $\lambda' = 330$
 $\lambda'' = 507$

 $\lambda' = 5$
 $\lambda'' = 115$

 $\lambda' = 0$
 $\lambda'' = 0$

 $\lambda' = 0$
 $\lambda'' = 0$

 $\lambda' = 5$
 $\lambda'' = 1000$

 $\lambda' = 347$
 $\lambda'' = 593$

Mínimos Quadrados

- O passo central do algoritmo é aproximar o fluxo ótico f entre dois quadros por uma combinação linear \tilde{f} de fluxos canônicos

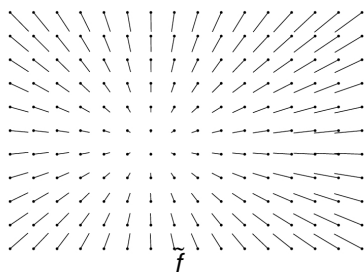
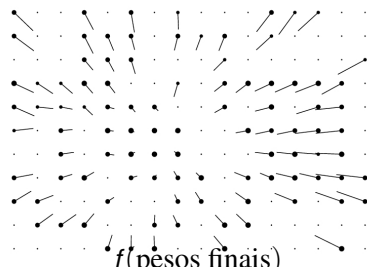
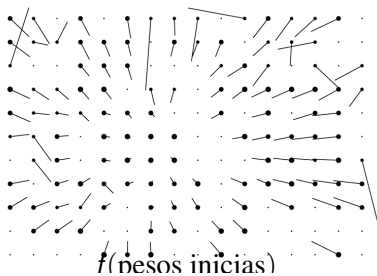
$$\tilde{f}(p) = \tilde{f}(R_x, R_y, R_z, R_f)(p) = R_x r_x(p) + R_y r_y(p) + R_z r_z(p) + R_f r_f(p)$$

- o produto escalar discreto de dois fluxos a e b e peso w é

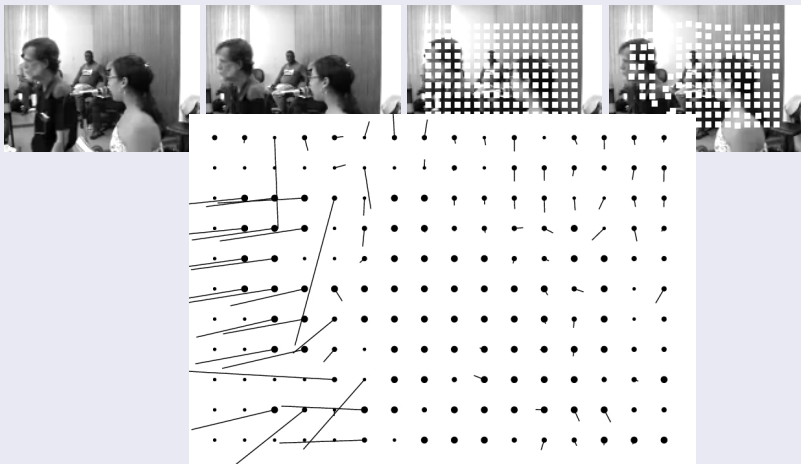
$$\langle\langle a|b \rangle\rangle = \frac{\sum_{i=1}^n w_i (a_i \cdot b_i)}{\sum_{i=1}^n w_i}$$

- O objetivo é determinar os valores de R_x, R_y, R_z e R_f que minimizam a discrepância $d = f - \tilde{f}(R_x, R_y, R_z, R_f)$ entre f e \tilde{f}

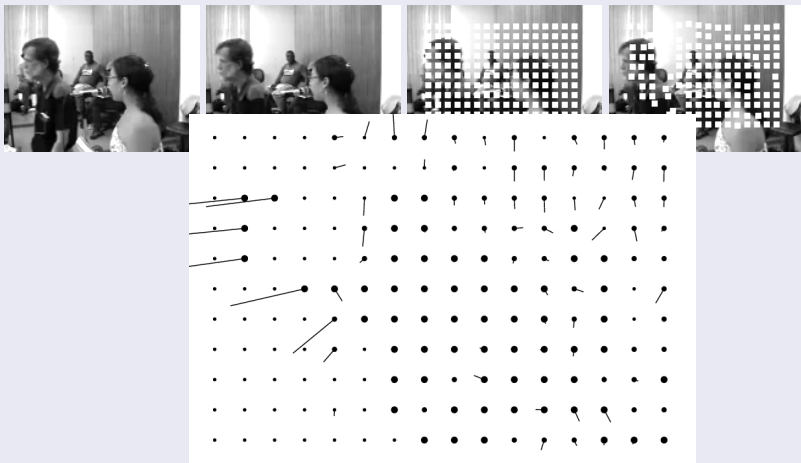
$$\begin{bmatrix} \langle\langle r_x|r_x \rangle\rangle & \langle\langle r_x|r_y \rangle\rangle & \langle\langle r_x|r_z \rangle\rangle & \langle\langle r_x|r_f \rangle\rangle \\ \langle\langle r_y|r_x \rangle\rangle & \langle\langle r_y|r_y \rangle\rangle & \langle\langle r_y|r_z \rangle\rangle & \langle\langle r_y|r_f \rangle\rangle \\ \langle\langle r_z|r_x \rangle\rangle & \langle\langle r_z|r_y \rangle\rangle & \langle\langle r_z|r_z \rangle\rangle & \langle\langle r_z|r_f \rangle\rangle \\ \langle\langle r_f|r_x \rangle\rangle & \langle\langle r_f|r_y \rangle\rangle & \langle\langle r_f|r_z \rangle\rangle & \langle\langle r_f|r_f \rangle\rangle \end{bmatrix} \begin{bmatrix} R_x \\ R_y \\ R_z \\ R_f \end{bmatrix} = \begin{bmatrix} \langle\langle f|r_x \rangle\rangle \\ \langle\langle f|r_y \rangle\rangle \\ \langle\langle f|r_z \rangle\rangle \\ \langle\langle f|r_f \rangle\rangle \end{bmatrix}$$



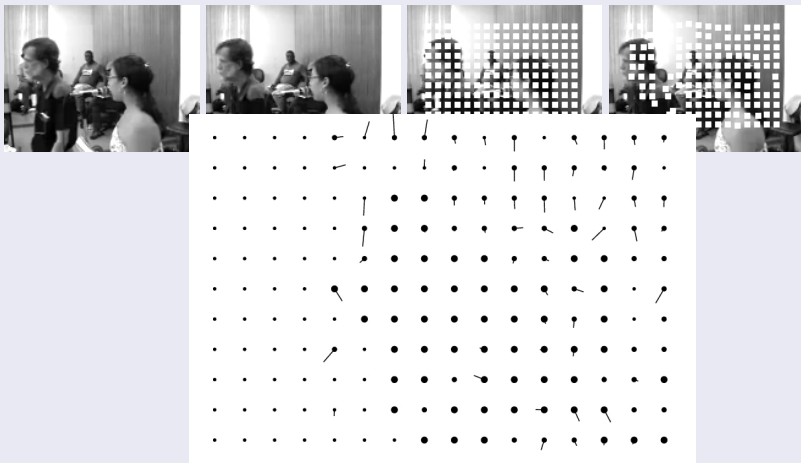
Ajuste iterativo dos pesos



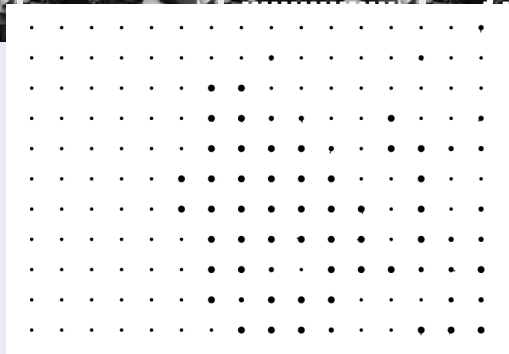
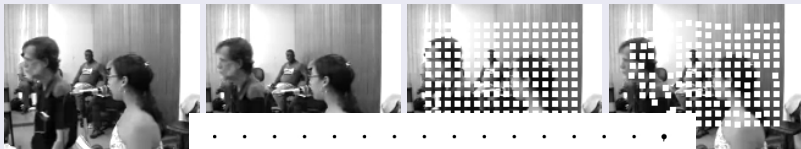
Ajuste iterativo dos pesos - 1 iteração



Ajuste iterativo dos pesos - 2 iteração



Ajuste iterativo dos pesos - 3 iteração



Medidas

- Medida de movimentação da câmera:

$$V = \sqrt{R_x + R_y + R_z + R_f}$$

- Magnitude da discrepância:

$$D = \|f - \tilde{f}(R_x, R_y, R_z, R_f)\|$$

Medidas

- Resíduo:

$$\tau \leftarrow \frac{\sqrt{\sum_{r \in p_i + \Omega} (J(r) - K(r + f_i))^2}}{|\Omega|}$$

Onde J e K são duas imagens próximas e Ω é uma janela.

