A PDE-based Level-Set Approach for Detection and Tracking of Moving Objects [1]

Authors: N. Paragois, R. Deriche,

Presenter: Egor Larionov

November 21, 2013

Outline

Introduction

Applications:

- Computer vision

Introduction

Applications:

- Computer vision
  - Surveillance (e.g. security, traffic monitoring, people counting)
  - Human computer interaction (e.g. Kinect)
  - Face detection (e.g. in digital cameras)
  - Automatic guidance (e.g. digital image stabilization)
  - Medical imaging (e.g. surgery assistance)
  - Animation (e.g. convert video to animation)

Introduction

Assume a “static” background:

Introduction

Assume a “static” background:

- Background in video is never strictly static. Why?

Introduction

Assume a "static" background:

- Background in video is never strictly static. Why?
  - Noise
  - Changes in illumination
  - Shadows

Introduction

Assume a “static” background:

- Background in video is never strictly static. Why?
  - Noise
  - Changes in illumination
  - Shadows

(http://gifbib.com/wp-content/uploads/2012/02/camera.gif)

Introduction

First approach:

Introduction

First approach:

- Take the difference (in intensity) of consecutive frames

Introduction

First approach:

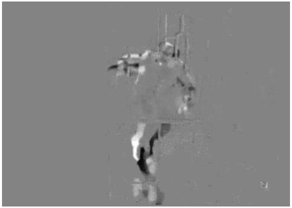
- Take the difference (in intensity) of consecutive frames
- Moving objects appear where the difference is large

Introduction

First approach:

- Take the difference (in intensity) of consecutive frames
- Moving objects appear where the difference is large

Example: (frame 23 – 22)

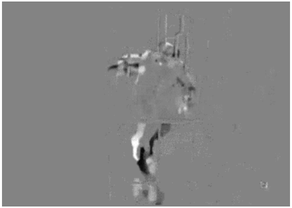


Introduction

First approach:

- Take the difference (in intensity) of consecutive frames
- Moving objects appear where the difference is large

Example: (frame 23 – 22)



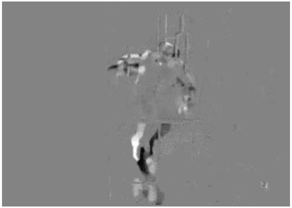
$$\xrightarrow{|\mathcal{I}-0.5|>0.15}$$

Introduction

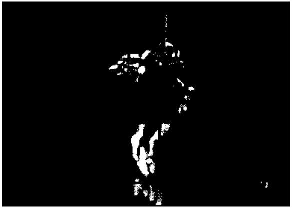
First approach:

- Take the difference (in intensity) of consecutive frames
- Moving objects appear where the difference is large

Example: (frame 23 – 22)



$$\xrightarrow{|\mathcal{I}-0.5|>0.15}$$

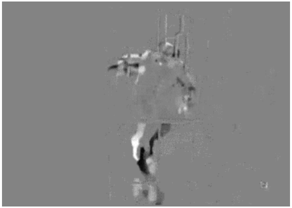


Introduction

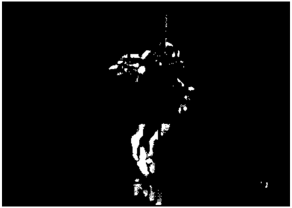
First approach:

- Take the difference (in intensity) of consecutive frames
- Moving objects appear where the difference is large

Example: (frame 23 – 22)



$$\xrightarrow{|\mathcal{I}-0.5|>0.15}$$



Problems:

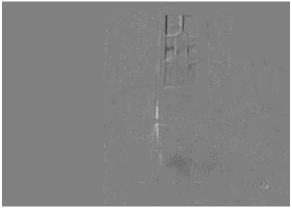
- Can't see the outline of the object
- Can't see temporarily stationary objects
- In general: detect too much or too little (threshold)

Introduction

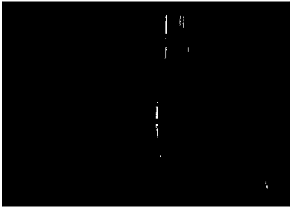
First approach:

- Take the difference (in intensity) of consecutive frames
- Moving objects appear where the difference is large

Example: (all frames)



$$\xrightarrow{|\mathcal{I}-0.5|>0.15}$$



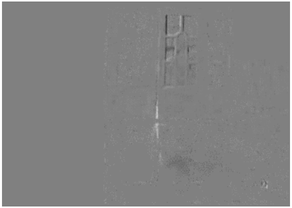
Problems:

- Can't see the outline of the object
- Can't see temporarily stationary objects
- In general: detect too much or too little (threshold)

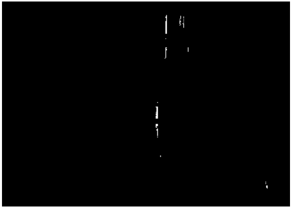
Introduction

Background Subtraction approach:

- Take the difference between background and each frame



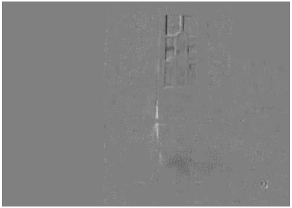
$$\xrightarrow{|\mathcal{I}-0.5|>0.15}$$



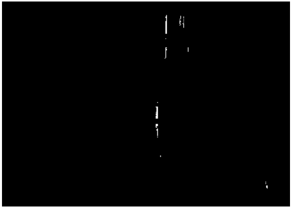
Introduction

Background Subtraction approach:

- Take the difference between background and each frame



$$\xrightarrow{|\mathcal{I}-0.5|>0.15}$$

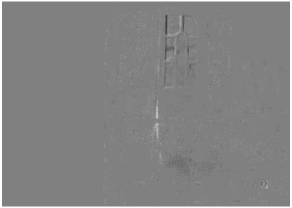


- Can now see temporarily stationary objects

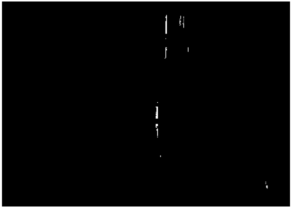
Introduction

Background Subtraction approach:

- Take the difference between background and each frame



$$\xrightarrow{|\mathcal{I}-0.5|>0.15}$$



- Can now see temporarily stationary objects
- However now we need a background image

Introduction

Other Background Subtraction approaches:

Introduction

Other Background Subtraction approaches:

- Adaptive Median Filtering (AMF) (1995) [2]
- Running Gaussian Average (1996) [3]
- Mixture of Gaussians (MoG) (2000) [4]
- Zivkovic AGMM (adaptive Gaussian mixtures) (2004) [5]
- Eigenbackgrounds (2000) [6]
- Prati Mediod (mediod filtering) (2003) [7]

Introduction

Other Background Subtraction approaches:

- Adaptive Median Filtering (AMF) (1995) [2]
- Running Gaussian Average (1996) [3]
- Mixture of Gaussians (MoG) (2000) [4]
- Zivkovic AGMM (adaptive Gaussian mixtures) (2004) [5]
- Eigenbackgrounds (2000) [6]
- Prati Mediod (mediod filtering) (2003) [7]

Most methods here are statistical in nature.

Author's Approach

Start with two initial frames:

Author's Approach

Start with two initial frames:

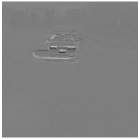
Motion detection:

Author's Approach

Start with two initial frames:

Motion detection:

Look at difference between consecutive frames:

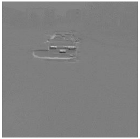


Author's Approach

Start with two initial frames:

Motion detection:

Look at difference between consecutive frames:



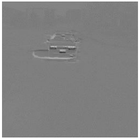
Build an image with large gradient at boundary:

Author's Approach

Start with two initial frames:

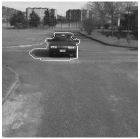
Motion detection:

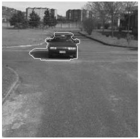
Look at difference between consecutive frames:



Build an image with large gradient at boundary:

Segmentation using Level Set:





Author's Approach

Tracking:

Author's Approach

Tracking:

- Use curve from before, continue segmentation on one of the original frames:

Note: in fact we evolve the curve to fit the original frame and the difference image simultaneously. Otherwise this method is same as Caselles and Coll [8]

Author's Approach

Main idea: do *motion detection* and *tracking* **simultaneously**.

Author's Approach

Main idea: do *motion detection* and *tracking* **simultaneously**.

- Builds on Caselles and Coll [8] for geodesic active contours

Author's Approach

Main idea: do *motion detection* and *tracking* **simultaneously**.

- Builds on Caselles and Coll [8] for geodesic active contours
- Builds on
  - Narrow Band [9] and
  - Fast Marching [10]

  front propagation methods

Author's Approach

Main idea: do *motion detection* and *tracking* **simultaneously**.

- Builds on Caselles and Coll [8] for geodesic active contours
- Builds on
  - Narrow Band [9] and
  - Fast Marching [10]

  front propagation methods
- Combines the two into a new front propagation method:
  - Hermes Algorithm

Author's Approach

Main idea: do *motion detection* and *tracking* **simultaneously**.

- Builds on Caselles and Coll [8] for geodesic active contours
- Builds on
  - Narrow Band [9] and
  - Fast Marching [10]

  front propagation methods
- Combines the two into a new front propagation method:
  - Hermes Algorithm

(We will focus on **building** the **Level Set PDE** (first bullet))

The Model

Recall **Geodesic Active Contours** formulation:

The Model

Recall **Geodesic Active Contours** formulation:

Image: $\mathcal{I}(x, y) : [0, a] \times [0, b] \to \mathbb{R}^+$

The Model

Recall **Geodesic Active Contours** formulation:

Image: $\mathcal{I}(x, y) : [0, a] \times [0, b] \to \mathbb{R}^+$

Curve: $\mathcal{C}(p) : [0, 1] \to \mathbb{R}^2$

The Model

Recall **Geodesic Active Contours** formulation:

Image:	$\mathcal{I}(x, y) : [0, a] \times [0, b] \rightarrow \mathbb{R}^+$
Curve:	$\mathcal{C}(p) : [0, 1] \rightarrow \mathbb{R}^2$
Normal to the curve:	$\vec{\mathcal{N}}(\mathcal{C}(p)) : \mathcal{C}([0, 1]) \rightarrow \mathbb{R}^2$

The Model

Recall **Geodesic Active Contours** formulation:

Image: $\mathcal{I}(x, y) : [0, a] \times [0, b] \to \mathbb{R}^+$

Curve: $\mathcal{C}(p) : [0, 1] \to \mathbb{R}^2$

Normal to the curve: $\vec{\mathcal{N}}(\mathcal{C}(p)) : \mathcal{C}([0, 1]) \to \mathbb{R}^2$

Implicit function: $u(x, y) : [0, a] \times [0, b] \to \mathbb{R}$ such that

$$\mathcal{C}([0, 1]) = \{(x, y) : u(x, y) = 0\} \quad \text{and} \quad \|\nabla u\| = \vec{\mathcal{N}}$$

The Model

Recall **Geodesic Active Contours** formulation:

Image: $\mathcal{I}(x,y) : [0,a] \times [0,b] \to \mathbb{R}^+$

Curve: $\mathcal{C}(p) : [0,1] \to \mathbb{R}^2$

Normal to the curve: $\vec{\mathcal{N}}(\mathcal{C}(p)) : \mathcal{C}([0,1]) \to \mathbb{R}^2$

Implicit function: $u(x,y) : [0,a] \times [0,b] \to \mathbb{R}$ such that

$$\mathcal{C}([0,1]) = \{(x,y) : u(x,y) = 0\} \quad \text{and} \quad \|\nabla u\| = \vec{\mathcal{N}}$$

Curvature: $\kappa = \nabla \cdot \left(\frac{\nabla u}{\|\nabla u\|} \right)$

The Model

Recall **Geodesic Active Contours** formulation:

Image: $\mathcal{I}(x, y) : [0, a] \times [0, b] \to \mathbb{R}^+$

Curve: $\mathcal{C}(p) : [0, 1] \to \mathbb{R}^2$

Normal to the curve: $\vec{\mathcal{N}}(\mathcal{C}(p)) : \mathcal{C}([0, 1]) \to \mathbb{R}^2$

Implicit function: $u(x, y) : [0, a] \times [0, b] \to \mathbb{R}$ such that

$$\mathcal{C}([0, 1]) = \{(x, y) : u(x, y) = 0\} \quad \text{and} \quad \|\nabla u\| = \vec{\mathcal{N}}$$

Curvature: $\kappa = \nabla \cdot \left(\frac{\nabla u}{\|\nabla u\|} \right)$

Goal: find $\mathcal{C}$ that minimizes

$$E(\mathcal{C}) = (1 - \lambda) \underbrace{\int_0^1 |\mathcal{C}'(p)|^2 dp}_{\text{internal}} + \lambda \underbrace{\int_0^1 g^2(|\nabla \mathcal{I}(\mathcal{C}(p))|) dp}_{\text{external}}$$

given some $\lambda \in [0, 1]$, and $g : \mathbb{R}^+ \to [0, 1]$ s.t. $g(r) \xrightarrow{r \to \infty} 0$, and $g(0) = 1$.

The Model

Need to modify the functional such that the curve approaches

- the boundary of motion (Motion Detection)

The Model

Need to modify the functional such that the curve approaches

- the boundary of motion (Motion Detection)
- the edges of the object (Tracking)

The Model

Need to modify the functional such that the curve approaches

- the boundary of motion (Motion Detection)
- the edges of the object (Tracking)

Idea: modify the image to have large gradient where the motion occurs!

Preparation

Define the inter-frame grey level difference image:

$$d(x, y) = \mathcal{I}(x, y; t + 1) - \mathcal{I}(x, y; t)$$

Let D be a r.v. with values from

$$\{d(x, y) : (x, y) \in [0, a] \times [0, b]\}.$$

Preparation

Define the inter-frame grey level difference image:

$$d(x, y) = \mathcal{I}(x, y; t + 1) - \mathcal{I}(x, y; t)$$

Let D be a r.v. with values from

$$\{d(x, y) : (x, y) \in [0, a] \times [0, b]\}.$$

Let L be a binary r.v. that can take values form $\{static,\ mobile\}$:

- $L = static$ (pixel is a background pixel)
- $L = mobile$ (pixel is on the moving object)

Probabilites

We define the probability density functions (pdf's):

- $p_{D|L}(d|static) \rightarrow$ pdf of observed d given a *static* pixel

Probabilites

We define the probability density functions (pdf's):

- $p_{D|L}(d|static) \rightarrow$ pdf of observed d given a *static* pixel
- $p_{D|L}(d|mobile) \rightarrow$ pdf of observed d given a *mobile* pixel

Probabilites

We define the probability density functions (pdf's):

- $p_{D|L}(d|static) \rightarrow$ pdf of observed d given a *static* pixel
- $p_{D|L}(d|mobile) \rightarrow$ pdf of observed d given a *mobile* pixel

A priori probabilities:

Probabilites

We define the probability density functions (pdf's):

- $p_{D|L}(d|static) \rightarrow$ pdf of observed d given a *static* pixel
- $p_{D|L}(d|mobile) \rightarrow$ pdf of observed d given a *mobile* pixel

A priori probabilities:

- $p_L(static) \rightarrow$ probability that a pixel is *static*

Probabilites

We define the probability density functions (pdf's):

- $p_{D|L}(d|static) \rightarrow$ pdf of observed d given a *static* pixel
- $p_{D|L}(d|mobile) \rightarrow$ pdf of observed d given a *mobile* pixel

A priori probabilities:

- $p_L(static) \rightarrow$ probability that a pixel is *static*
- $p_L(mobile) \rightarrow$ probability that a pixel is *mobile*

Probabilites

We define the probability density functions (pdf's):

- $p_{D|L}(d|static) \rightarrow$ pdf of observed d given a *static* pixel
- $p_{D|L}(d|mobile) \rightarrow$ pdf of observed d given a *mobile* pixel

A priori probabilities:

- $p_L(static) \rightarrow$ probability that a pixel is *static*
- $p_L(mobile) \rightarrow$ probability that a pixel is *mobile*

Now take a pixel (x, y) from some observed D.

Probabilites

We define the probability density functions (pdf's):

- $p_{D|L}(d|static) \rightarrow$ pdf of observed d given a *static* pixel
- $p_{D|L}(d|mobile) \rightarrow$ pdf of observed d given a *mobile* pixel

A priori probabilities:

- $p_L(static) \rightarrow$ probability that a pixel is *static*
- $p_L(mobile) \rightarrow$ probability that a pixel is *mobile*

Now take a pixel (x, y) from some observed D.
Then the probability this pixel has intensity d is given by:

$$p_D(d) = p_L(static) p_{D|static}(d|static) + p_L(mobile) p_{D|static}(d|mobile)$$

Follows from Bayes rule. Sometimes called marginalization.

Mixture of Distributions

- p_D is a mixture of Laplacian distributions, so:

$$p_{D|L}(d|\ell) = \frac{\lambda_\ell}{2} e^{-\lambda_\ell |d|}$$

Mixture of Distributions

- p_D is a mixture of Laplacian distributions, so:

$$p_{D|L}(d|\ell) = \frac{\lambda_\ell}{2} e^{-\lambda_\ell |d|}$$

- We have p_D, since we can observe it.

Mixture of Distributions

- p_D is a mixture of Laplacian distributions, so:

$$p_{D|L}(d|\ell) = \frac{\lambda_\ell}{2} e^{-\lambda_\ell |d|}$$

- We have p_D, since we can observe it.
- Need to find $p_L(\ell)$ and λ_ℓ.

Mixture of Distributions

- p_D is a mixture of Laplacian distributions, so:

$$p_{D|L}(d|\ell) = \frac{\lambda_\ell}{2} e^{-\lambda_\ell |d|}$$

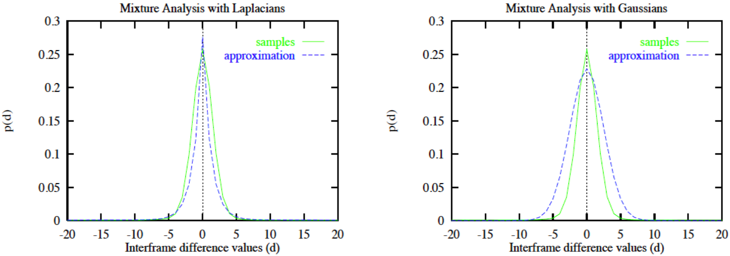
- We have p_D, since we can observe it.
- Need to find $p_L(\ell)$ and λ_ℓ.
- Use a Maximum Likelihood Estimation method [11].

Mixture of Distributions

- p_D is a mixture of Laplacian distributions, so:

$$p_{D|L}(d|\ell) = \frac{\lambda_\ell}{2} e^{-\lambda_\ell |d|}$$

- We have p_D, since we can observe it.
- Need to find $p_L(\ell)$ and λ_ℓ.
- Use a Maximum Likelihood Estimation method [11].



- Note: could use Gaussian distributions instead.

More probabilities!

Define two more probabilities (call them energies):

More probabilities!

Define two more probabilities (call them energies):

$$
\begin{aligned}
E_{trans}((x,y),(v,w)) &= p(d(x,y)|static)\cdot p(d(v,w)|mobile) \\
&\quad + p(d(x,y)|mobile)\cdot p(d(v,w)|static) \\
E_{smooth}((x,y),(v,w)) &= p(d(x,y)|static)\cdot p(d(v,w)|static) \\
&\quad + p(d(x,y)|mobile)\cdot p(d(v,w)|mobile)
\end{aligned}
$$

More probabilities!

Define two more probabilities (call them energies):

$$
\begin{aligned}
E_{trans}((x,y),(v,w)) &= p(d(x,y)|static) \cdot p(d(v,w)|mobile) \\
&\quad + p(d(x,y)|mobile) \cdot p(d(v,w)|static) \\
E_{smooth}((x,y),(v,w)) &= p(d(x,y)|static) \cdot p(d(v,w)|static) \\
&\quad + p(d(x,y)|mobile) \cdot p(d(v,w)|mobile)
\end{aligned}
$$

Interpret as probabilities for pixels (x,y) and (v,w) to:

$E_{trans}((x,y),(v,w))$

mark the boundary

$E_{smooth}((x,y),(v,w))$

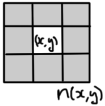
be on the same side

Detection

Finally define the new image:

$$\mathcal{I}_{detect}(x, y) = \max_{(v,w) \in n(x,y)} \left\{ \frac{E_{trans}((x, y), (v, w))}{E_{smooth}((x, y), (v, w))} \right\}$$

where $n(x, y)$ is the neighbourhood of the pixel (x, y):

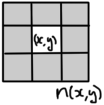


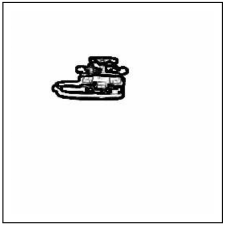
Detection

Finally define the new image:

$$\mathcal{I}_{detect}(x,y) = \max_{(v,w)\in n(x,y)} \left\{ \frac{E_{trans}((x,y),(v,w))}{E_{smooth}((x,y),(v,w))} \right\}$$

where $n(x,y)$ is the neighbourhood of the pixel (x,y):



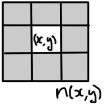


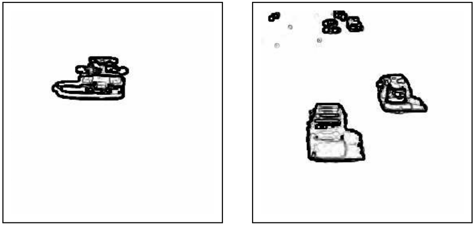
Detection

Finally define the new image:

$$\mathcal{I}_{detect}(x, y) = \max_{(v,w)\in n(x,y)} \left\{ \frac{E_{trans}((x, y), (v, w))}{E_{smooth}((x, y), (v, w))} \right\}$$

where $n(x, y)$ is the neighbourhood of the pixel (x, y):





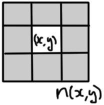
- Gives high gradient at the boundary of moving object

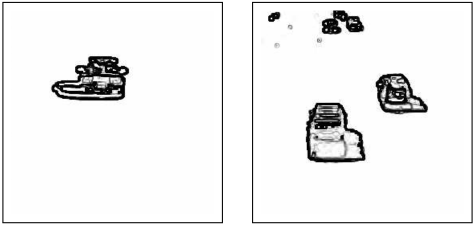
Detection

Finally define the new image:

$$\mathcal{I}_{detect}(x,y) = \max_{(v,w)\in n(x,y)} \left\{ \frac{E_{trans}((x,y),(v,w))}{E_{smooth}((x,y),(v,w))} \right\}$$

where $n(x,y)$ is the neighbourhood of the pixel (x,y):





- Gives high gradient at the boundary of moving object
- May apply smoothing, to reduce noise

Detection

Now modify the energy functional:

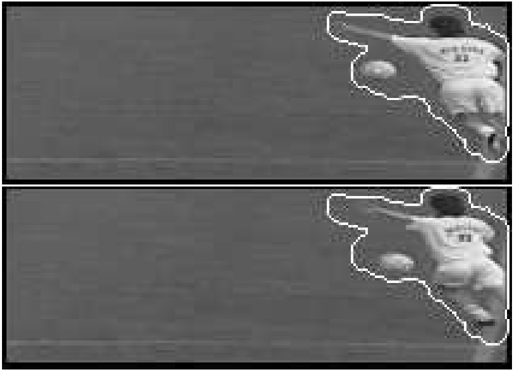
$$E(\mathcal{C}) = (1-\lambda)\int_0^1 |\mathcal{C}'(p)|^2 dp \; + \; \lambda\int_0^1 g^2(|\nabla \mathcal{I}_{detect}(\mathcal{C}(p))|)dp$$

Detection

Now modify the energy functional:

$$E(\mathcal{C}) = (1-\lambda)\int_0^1 |\mathcal{C}'(p)|^2 dp \; + \; \lambda \int_0^1 g^2(|\nabla \mathcal{I}_{detect}(\mathcal{C}(p))|) dp$$

An example of motion detection only:



Tracking

Finally need to insure curve segments the object in each frame. Modify the external part of the energy functional to be:

$$\int_0^1 \Big(\gamma \underbrace{g(|\nabla \mathcal{I}_{detect}(\mathcal{C}(p))|)}_{detection} + (1-\gamma) \underbrace{g(|\nabla \mathcal{I}_t(\mathcal{C}(p))|)}_{tracking} \Big)^2 dp$$

Tracking

Finally need to insure curve segments the object in each frame. Modify the external part of the energy functional to be:

$$\int_0^1 \left(\gamma \underbrace{g(|\nabla \mathcal{I}_{detect}(\mathcal{C}(p))|)}_{detection} + (1-\gamma) \underbrace{g(|\nabla \mathcal{I}_t(\mathcal{C}(p))|)}_{tracking} \right)^2 dp$$

where $\gamma \in [0,1]$ is some parameter that balances detection and tracking:

- $\gamma = 0 \implies$ usual geodesic active contour model
- $\gamma = 1 \implies$ only motion detection

Tracking

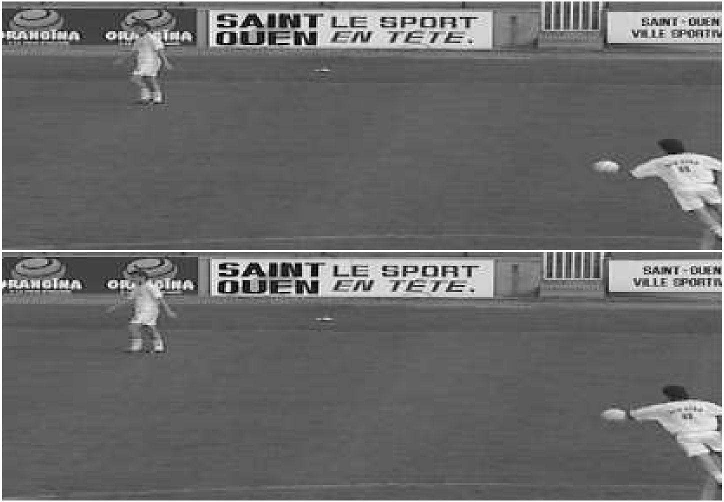
Solve as a geodesic active contour problem, to get

$$\begin{aligned}\frac{du}{dt} = \Bigg[\gamma\Bigg(&g(|\nabla \mathcal{I}_{detect}(C(p,t))|)\cdot\kappa(p,t)\\&+\nabla g(|\nabla \mathcal{I}_{detect}(C(p,t))|)\cdot\frac{\nabla u}{|\nabla u|}\Bigg)\\+(1-\gamma)\Bigg(&g(|\nabla \mathcal{I}_{t}(C(p,t))|)\cdot\kappa(p,t)\\&+\nabla g(|\nabla \mathcal{I}_{t}(C(p,t))|)\cdot\frac{\nabla u}{|\nabla u|}\Bigg)\Bigg]|\nabla u|,\end{aligned}$$

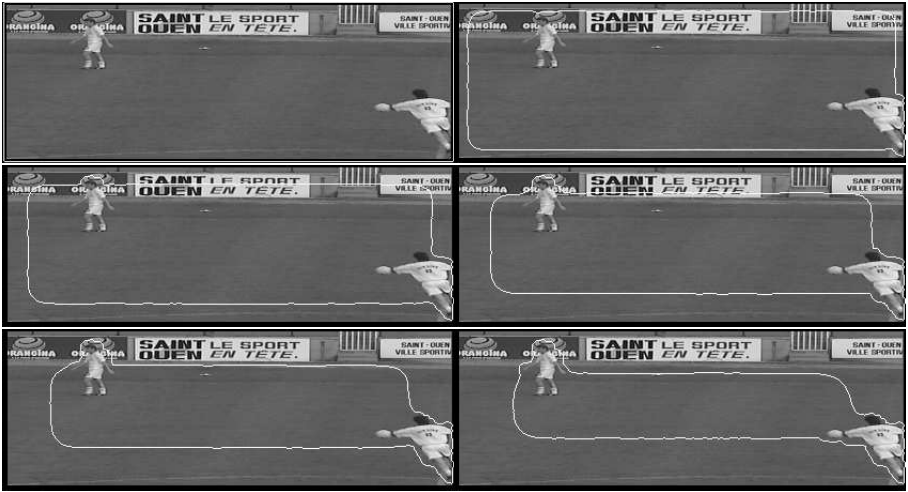
the speed function of the contour.

Numerical Examples: Football (Detection only)

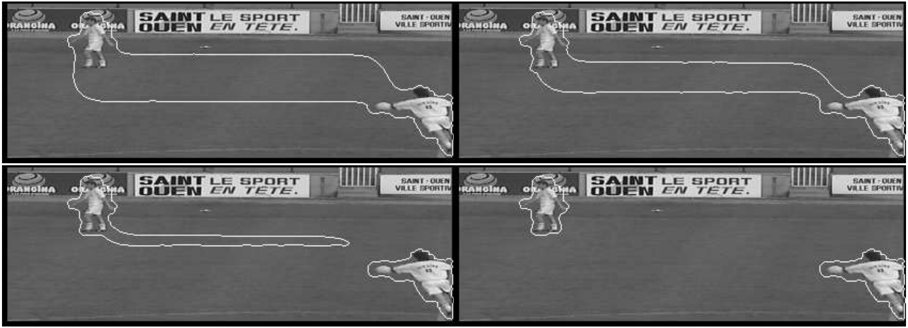
Football sequence. Two initial frames:



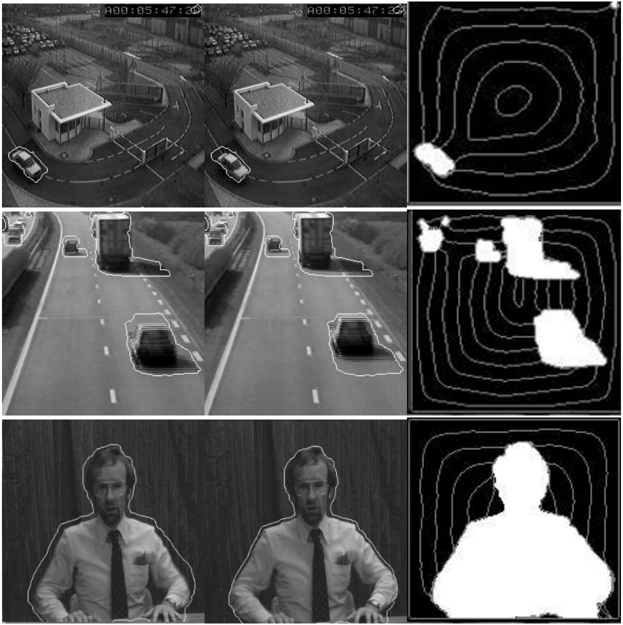
Numerical Examples: Football (Detection only)



Numerical Examples: Football (Detection only)

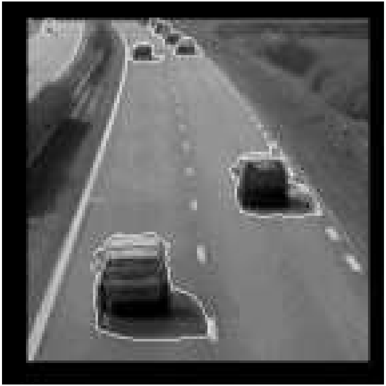


Numerical Examples: Various (Detection only)

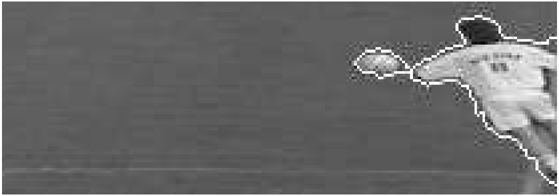


Numerical Examples: Highway (Detection only)

Numerical Examples: Highway (Detection and Tracking)



Numerical Examples: Football (Detection and Tracking)



Thank You!

The End

Bibliography I

N. K. Paragios and R. Deriche.
A pde-based level-set approach for detection and tracking of moving objects.
In *Computer Vision, 1998. Sixth International Conference on*, pp. 1139–1145. IEEE (1998).

N. J. McFarlane and C. P. Schofield.
Segmentation and tracking of piglets in images.
Machine Vision and Applications, 8(3), pp. 187–193 (1995).

C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland.
Pfinder: Real-time tracking of the human body.
Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(7), pp. 780–785 (1997).

Bibliography II

- C. Stauffer and W. E. L. Grimson.
  Learning patterns of activity using real-time tracking.
  Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(8), pp. 747–757 (2000).

- Z. Zivkovic.
  Improved adaptive gaussian mixture model for background subtraction.
  In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, pp. 28–31. IEEE (2004).

- N. M. Oliver, B. Rosario, and A. P. Pentland.
  A bayesian computer vision system for modeling human interactions.
  Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(8), pp. 831–843 (2000).

Bibliography III

- R. Cucchiara, C. Grana, M. Piccardi, and A. Prati.
  Detecting moving objects, ghosts, and shadows in video streams.
  Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25(10), pp. 1337–1342 (2003).

- V. Caselles and B. Coll.
  Snakes in movement.
  SIAM Journal on Numerical Analysis, 33(6), pp. 2445–2456 (1996).

- D. Adalsteinsson.
  A fast level set method for propagating interfaces.
  Ph.D. thesis, Citeseer (1994).

Bibliography IV

J. A. Sethian.
A fast marching level set method for monotonically advancing fronts.
Proceedings of the National Academy of Sciences, 93(4), pp. 1591–1595 (1996).

N. Paragios and C. Tziritas.
Detection and location of moving objects using deterministic relaxation algorithms.
In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 1, pp. 201–205. IEEE (1996).