Aligning Sequences and Actions by Maximizing Space-Time Correlations

Yaron Ukrainitz and Michal Irani

Presented by, Deborah Goshorn

CSE 252C – Dr. Serge Belongie

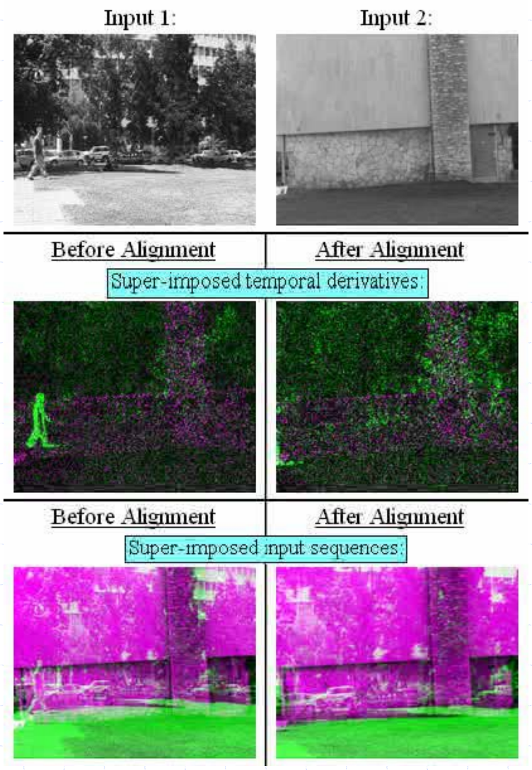
Outline

- Introduction
- Image Alignment first!
- Algorithm
- Experimental Results
- More research...

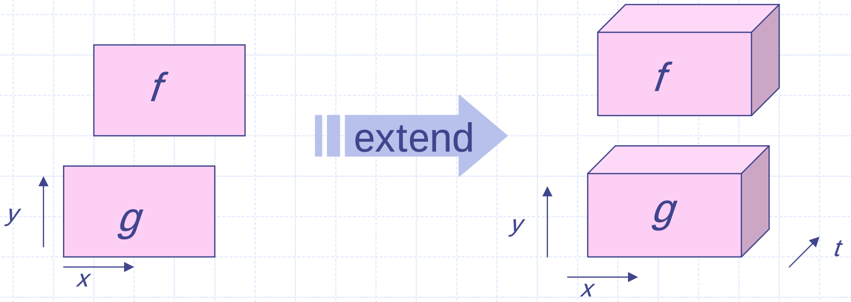
Introduction

Sequence alignment – wide range of scenarios:

- **Align sequence pairs with:**
  - * stationary/(jointly moving) cameras
  - * same/different photometric properties
  - * with/without moving objects
- **Algorithm applies to intensity information**
  - * Without segmenting foreground
  - * Without a priori finding corresponding features across sequences



Main Goal:



Extend previous research of **image** alignment to **space-time** alignment

Previously….

Multi-Sensor Image Alignment

Michal Irani & P. Anandan

Identify an ***image* representation** for multi-sensor alignment

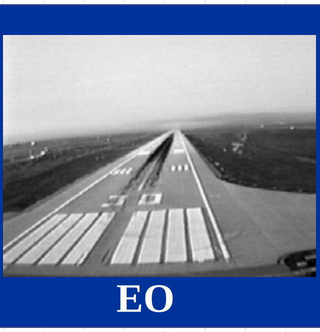
- does **not** rely on **sparse image features** *(e.g. edge, contour, point features)*

Presents new **alignment** technique

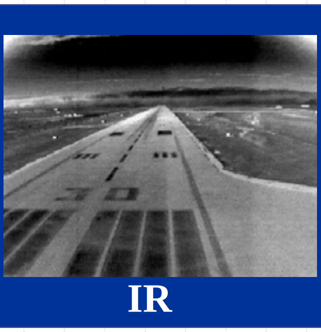
- applies global estimation to any choice of **local similarity** measure

Problems

1. What is the **relationship** between the **brightness** values of pixels in an image from sensor 1 and in another image from sensor 2, of **different modality**?
2. **Contrast reversal** may occur between the 2 images in **some** places of the image but not in others



EO

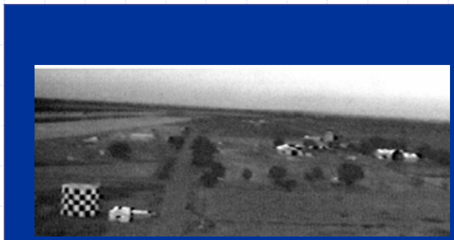


IR

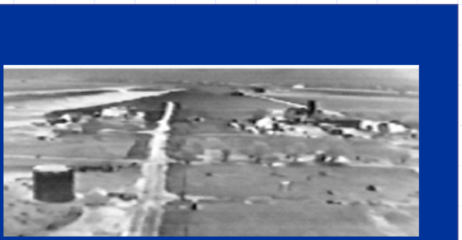
Problems

3. Visual features present in one image but not the other (**mutually exclusive features**)

4. **Multiple** brightness values in one image may map to one **single** brightness value in other image, vice versa.



EO



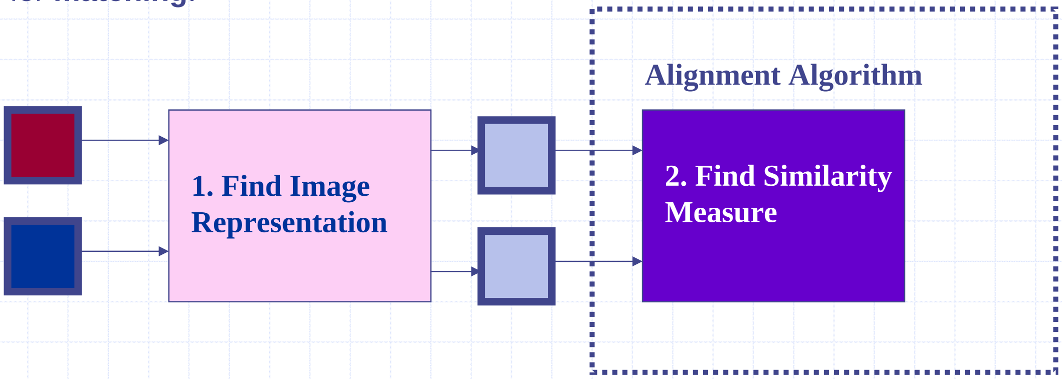
IR

In other words…

The 2 images are usually not globally correlated!

… So, one thinks to find:

1. A good **image representation** to work with, which brings out **common** information between the 2 images and also suppresses **non-common** information
2. Once we found the image representation, now find the right **similarity** measure for **matching**.



Previous Work:

1. Methods that use **invariant image representation**.

   ex.) edge maps, oriented edge vector fields, contour features, feature points.

   Information loss, because of **thresholding** steps

   Sparse set of highly significant features

   Threshold choice is very **data** and **sensor dependent**

2. Methods that use **invariant similarity measure** to register multi-sensor images.

   ex) **mutual information**, proposed method

Alignment by Maximization of Mutual Information – Viola & Wells (1997)

- Also did intensity based not feature based
- Efficient because uses stochastic approximation (noisy derivatives in gradient descent algorithm)
- Claims mutual information is more robust than traditional **correlation**

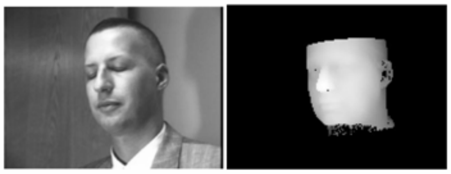


Figure 1: Two different views of RK. On the left is a video image. On the right is a depth map of a model of RK that describes the distance to each of the visible points of the model. Closer points are rendered brighter than more distant ones.

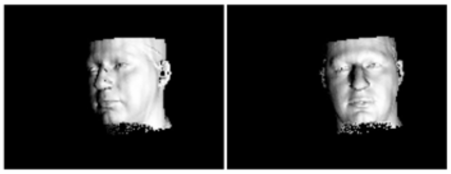


Figure 2: At left is a rendering of a 3D model of RK. The position of the model is the same as the position of the actual head. At right is a rendering of the head model in an incorrect pose.

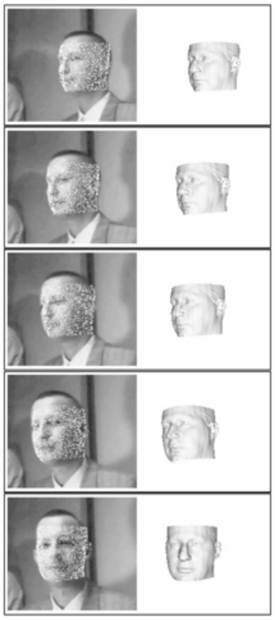
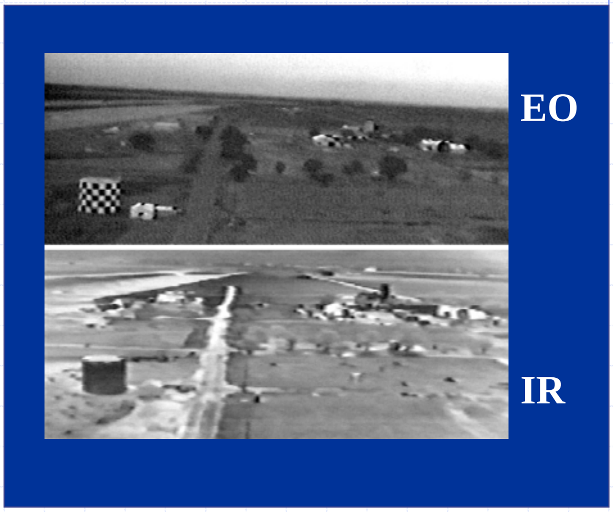


Figure 10: Video Head Tracking Experiment

Why not the mutual information method?

Authors claim the mutual information method:

1. Assumes the 2 images have a **global** stat. correlation (violated)
2. Since stat. correlation between raw multi-sensor images **decreases** as spatial resolution **decreases**, it will **not** extend to **coarse-to-fine** estimation, (which is often used to fix large misalignments)

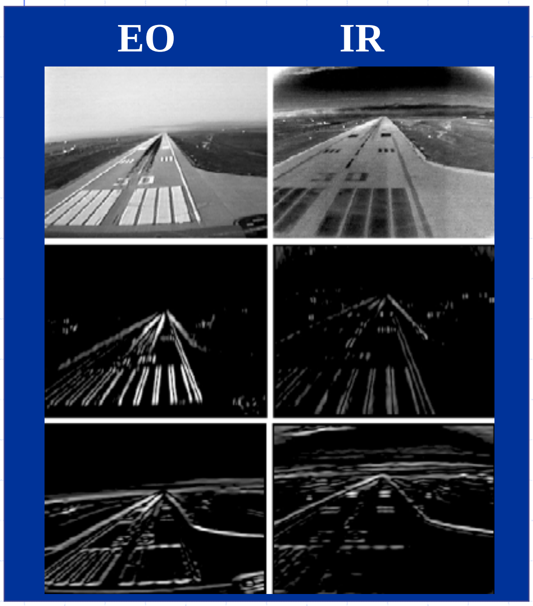


How they handled the problems:

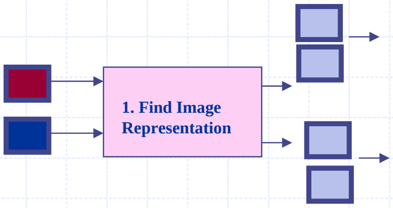
1. Only assuming a **local** correlation, not global, of the images.
2. Method is invariant to **contrast reversal**
3. Method provides **orientational** sensitivity
4. Method suitable for **coarse-to-fine** processing
5. Method rejects **outliers** (I.e. mutually exclusive visual features)

Image Representation

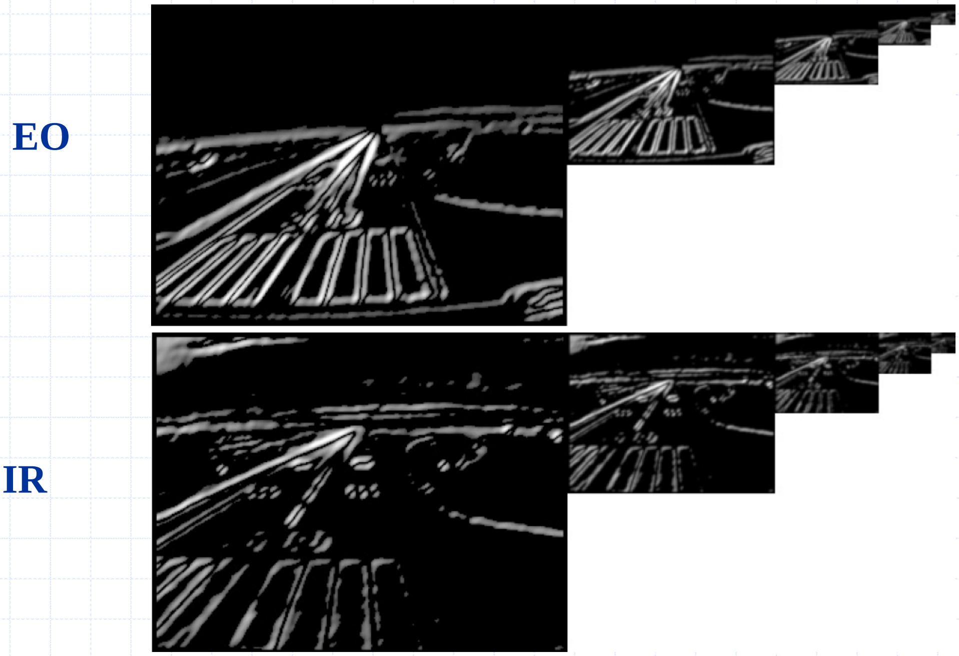
At low resolution levels, we must still capture small (high-resolution) temporal changes!



… Apply **directional derivative filters**, then **square** it (to handle contrast reversal)


1. Find Image Representation

Since we handled coarse-to-fine, we can fix large misalignments by constructing Gaussian pyramid!



Global Alignment

Estimate parametric transformation **globally**

- Useful due to the plurality of outliers across sensors and hence the unreliability of local matches**.**

Global estimation applied directly to **local correlation functions**

-could have used **local mutual information** here

Normalized-Correlation as a Local Similarity Measure

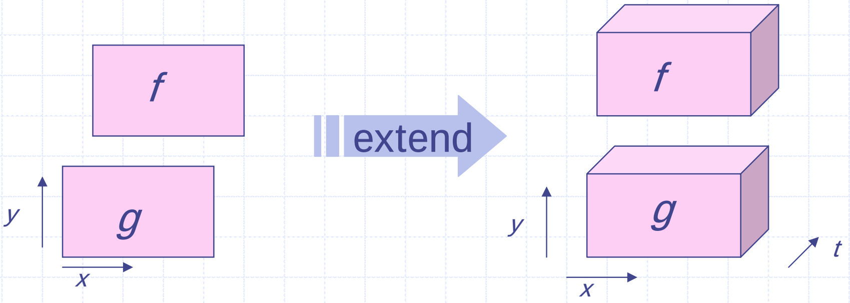
- Invariant to local changes in mean and contrast

- Locally, within small image patches which contain corresponding image features, stat. correlation is high

- Normalized-correlation is a linear approx to stat. correlation of 2 signals in a small window – cheaper to compute!

$$NC(w_f, w_g) = \frac{\mathrm{cov}(w_f, w_g)}{\sqrt{\mathrm{var}(w_f)}\sqrt{\mathrm{var}(w_g)}}$$

Extend image alignment to space-time alignment!



Definitions

Sequence alignment: “finding the spatial and temporal coordinate transformation that bring one sequence into alignment with the other, both in **space** and **time**”

* alignment *not* recognition

Action alignment: recover space-time alignment transformations in sequences when the **same** action is performed at different times/places/people/(sensors)/(speeds)

Multi-Sensor Alignment: sequences are simultaneous records of of the same scene, recorded using different **sensor modalities**

Introduction

•Aligns sequences of **same action**

either:

***** at different times/places/people/people
(different speeds)

or:

***** same scene but multiple cameras
(different modalities)

Observations

1. Temporal changes are captured in space-time volumes created by 2 sequences, ***not*** individual frames

→ **Sequence to Sequence** Alignment is “better” than **Image to Image** Alignment

Problem Formulation

$f, g:$ The sequences (or filtered versions) to align.

$\mathbf{p}:$ The spatio-temporal parametric transformation vector that maximizes...

$M(f, g):$ A global similarity measure btw *f*, *g* after aligning *f* and *g*

Problem Formulation

(x, y, t): One **space-time point** in a sequence.

$\mathbf{u} := (u_1, u_2, u_3)$ Spatio-temporal **displacement** vector

$$= u(x, y; \mathbf{p}) = \begin{bmatrix} u_1(x, y, t; \mathbf{p}) \\ u_2(x, y, t; \mathbf{p}) \\ u_3(x, y, t; \mathbf{p}) \end{bmatrix} = \begin{bmatrix} p_1 x + p_2 y + p_3 \\ p_4 x + p_5 y + p_6 \\ p_7 t + p_8 \end{bmatrix}$$

•**1-D** affine transformation for **time**

•**2-D** affine transformation for **space** (okay because assuming planar, i.e. distant, or the 2 cameras are close to each other)

Alignment Algorithm

Algorithm:

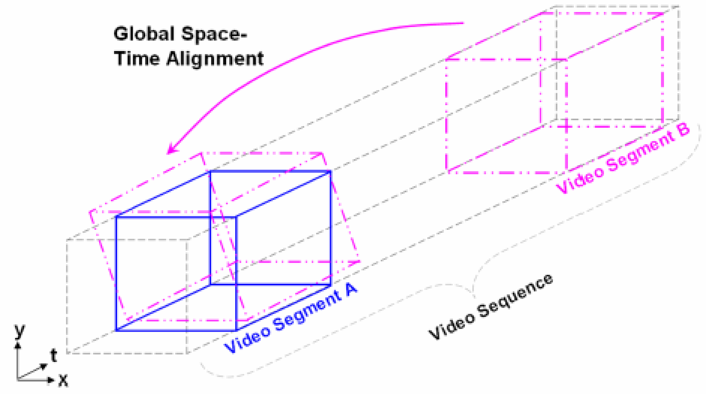


Figure 1: **Global Space-Time Alignment**
Movie segment B in the video sequence is space-time aligned with movie segment A using a space-time volumetric parametric transformation.

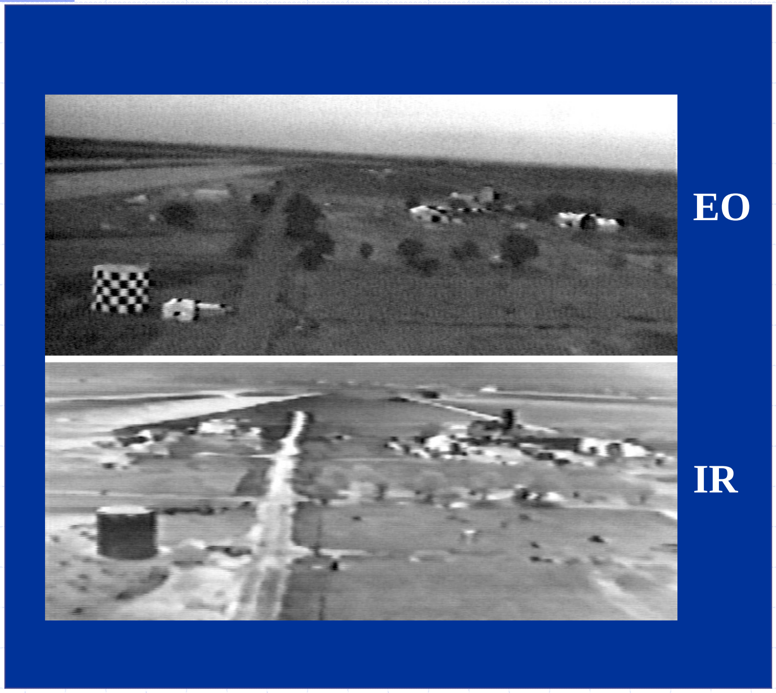
Alignment Algorithm

Algorithm:

```
1. Make a space-time gaussian pyramid for each sequence
2. Find initial guess $p_o$
3. Apply maximization iterations in the current pyramid level until convergence
      - use current parameter estimate $p_o$ from the last iteration to find delta
      - Update current parameter estimate $p_o^* = p_o$ + delta
      - Test for convergence: if $\boxed{M(p_o) - M(p_o^*)} < eps$ → go to 3, else break
4. Go to next pyramid level & then go to 3
```

Similarity Measure M(·)

Outliers!



Outliers

-For corresponding space-time blocks that have **mutually exclusive** image features, the normalized correlation function is not concave shape.

- Thus, we count those small space time blocks as outliers.

-To measure concavity, take determinant of Hessian

$$H_{C(x,y,t)} = \begin{bmatrix} \frac{\partial^2 C^{(x,y,t)}}{\partial x^2} & \frac{\partial^2 C^{(x,y,t)}}{\partial x \partial y} & \frac{\partial^2 C^{(x,y,t)}}{\partial x \partial t} \\ \frac{\partial^2 C^{(x,y,t)}}{\partial y \partial x} & \frac{\partial^2 C^{(x,y,t)}}{\partial y^2} & \frac{\partial^2 C^{(x,y,t)}}{\partial y \partial t} \\ \frac{\partial^2 C^{(x,y,t)}}{\partial t \partial x} & \frac{\partial^2 C^{(x,y,t)}}{\partial t \partial y} & \frac{\partial^2 C^{(x,y,t)}}{\partial t^2} \end{bmatrix}$$

Similarity Measure .Similarity Measure M(.)

IMAGES

-Globally, intensities of two **images** have non-linear transformations (depends on intensity & location)

➔ can't use **mutual information**

-Need a a **local similarity** measure (for small corresponding (space) image patches) that is invariant to **linear** intensity transformations!

➔ can use **normalized correlation**

Image & Time Warping

- To improve performance, warp each space-time block before each iteration.

- Warp space-time block toward reference space time block according to the current estimated parametric transformation $\mathbf{p}_0$

- Compensates for spatial distortions between pairs of sequences

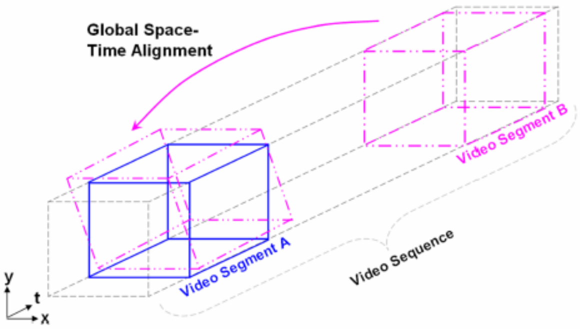
- Improves quality of correlation

Similarity Measure

Extend IMAGES to VIDEO SEQUENCES

w_f, w_g = corresponding space-time patches/windows

- Compute local normalized correlations within these small space-time patches (e.g. 7x7x7)



Alignment Algorithm

What do we maximize?

For each space-time patch, compute normalized correlation:

$$NC(w_f, w_g) = \frac{\mathrm{cov}(w_f, w_g)}{\sqrt{\mathrm{var}(w_f)}\sqrt{\mathrm{var}(w_g)}}$$

squared normalized correlation to mitigate contrast reversal

$$C(w_f, w_g) = \frac{\mathrm{cov}^2(w_f, w_g)}{\mathrm{var}(w_f)\mathrm{var}(w_g) + \alpha}$$

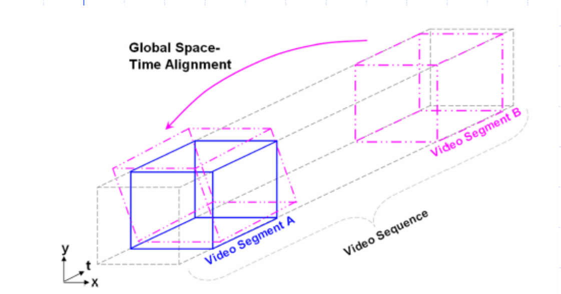
sum squared normalized correlation over all dimensions

$$M(f,g) = \sum_x \sum_y \sum_t \boxed{C\left(w_f(x,y,t), w_g(x,y,t)\right)}$$

Rewrite C() in terms of displacement vector **u**...

$$C^{(x,y,t)}(\boldsymbol{u}) = C\left(w_f(x,y,t), w_g(x+u_1, y+u_2, t+u_3)\right)$$

...so that we can rewrite similarity measure M() in terms of parameter vector **p**



We find the **p** that maximizes this!

$$M(\boldsymbol{p}) = \sum_{(x,y,t)\in f} C^{(x,y,t)}\left(\boldsymbol{u}(x,y,t;\boldsymbol{p})\right)$$

Alignment Algorithm

The Maximization Process

Algorithm:

1. Make a space-time Gaussian pyramid for each sequence
2. Find initial guess p_o (at lowest resolution)
3. Apply maximization iterations in the current pyramid level until convergence
   - **use current parameter estimate p_o from the last iteration to find delta**
   - Update current parameter estimate $p_o^* = p_o + \boxed{\text{delta}}$
   - Test for convergence: if $M(p_o) - M(p_o^*) <$ eps go to 3, else break
4. Apply maximization iterations in the current pyramid level until convergence, go up to next pyramid level (the next finest resolution).

How much to warp in space and time!

Alignment Algorithm

The Maximization Process

Algorithm:

1. Make a space-time Gaussian pyramid for each sequence
2. Find initial guess p_o (at lowest resolution)
3. Apply maximization iterations in the current pyramid level until convergence
   - **use current parameter estimate p_o from the last iteration to find delta**
   - Update current parameter estimate $p_o^* = p_o + \boxed{\text{delta}}$
   - Test for convergence: if $M(p_o) - M(p_o^*) < eps$ → go to 3, else break
4. Apply maximization iterations in the current pyramid level until convergence, go up to next pyramid level (the next finest resolution).

$$\mathbf{u} := (u_1, u_2, u_3) = u(x,y;\mathbf{p}) = \begin{bmatrix} u_1(x,y,t;\mathbf{p}) \\ u_2(x,y,t;\mathbf{p}) \\ u_3(x,y,t;\mathbf{p}) \end{bmatrix} = \begin{bmatrix} p_1x + p_2y + p_3 \\ p_4x + p_5y + p_6 \\ p_7t + p_8 \end{bmatrix}$$

How much to warp in space and time!

Rewrite math in matrix form...

$$\mathbf{u} := (u_1, u_2, u_3) = u(x,y;\mathbf{p}) = \begin{bmatrix} u_1(x,y,t;\mathbf{p}) \\ u_2(x,y,t;\mathbf{p}) \\ u_3(x,y,t;\mathbf{p}) \end{bmatrix} = \begin{bmatrix} p_1x + p_2y + p_3 \\ p_4x + p_5y + p_6 \\ p_7t + p_8 \end{bmatrix}$$

...as

$$\boldsymbol{u}(x,y,t;\boldsymbol{p}) = X(x,y,t) \cdot \boldsymbol{p}$$

where $X(x,y,t) = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t & 1 \end{bmatrix}.$ & $\boldsymbol{p} = (p_1, \ldots, p_8)$

Newton's Method to Maximize Similarity Measure M()

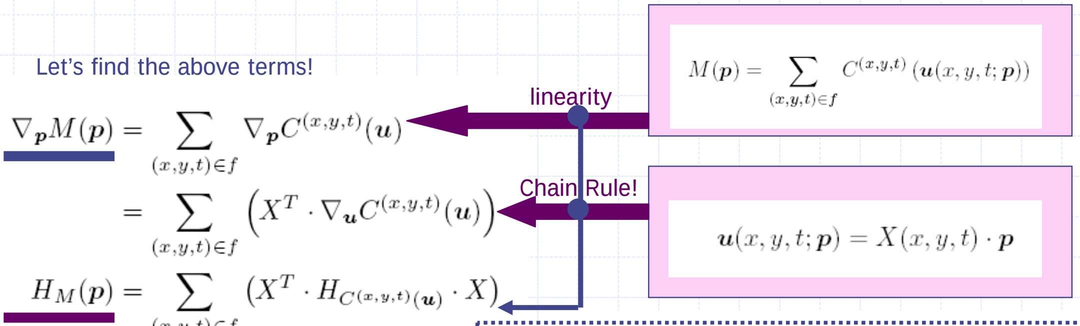
To perform Newton's Method, approximate M() with 2nd order Taylor Series Expansion:

$$M(\boldsymbol{p}) = M(\boldsymbol{p}_0) + \left(\underline{\nabla_{\boldsymbol{p}} M(\boldsymbol{p}_0)}\right)^T \boldsymbol{\delta}_p + \frac{1}{2}\boldsymbol{\delta}_p^T \underline{H_M(\boldsymbol{p}_0)}\boldsymbol{\delta}_p$$

To find optimal **p**, iteratively increment p with step-size given by the formula:

$$\boldsymbol{\delta}_p = -\left(H_M(\boldsymbol{p}_0)\right)^{-1} \cdot \nabla_{\boldsymbol{p}} M(\boldsymbol{p}_0)$$

Let's find the above terms!



$$M(\boldsymbol{p}) = \sum_{(x,y,t)\in f} C^{(x,y,t)}\left(\boldsymbol{u}(x,y,t;\boldsymbol{p})\right)$$

$$\boldsymbol{u}(x,y,t;\boldsymbol{p}) = X(x,y,t)\cdot\boldsymbol{p}$$

linearity

Chain Rule!

$$\underline{\nabla_{\boldsymbol{p}} M(\boldsymbol{p})} = \sum_{(x,y,t)\in f} \nabla_{\boldsymbol{p}} C^{(x,y,t)}(\boldsymbol{u})$$

$$= \sum_{(x,y,t)\in f} \left(X^T \cdot \nabla_{\boldsymbol{u}} C^{(x,y,t)}(\boldsymbol{u})\right)$$

$$\underline{H_M(\boldsymbol{p})} = \sum_{(x,y,t)\in f} \left(X^T \cdot H_{C^{(x,y,t)}(\boldsymbol{u})} \cdot X\right)$$

Note: $\nabla_{\boldsymbol{u}} C^{(x,y,t)} = \left[\frac{\partial C^{(x,y,t)}}{\partial x} \ \frac{\partial C^{(x,y,t)}}{\partial y} \ \frac{\partial C^{(x,y,t)}}{\partial t}\right]^T$

Alignment Algorithm

how much to warp!?

Rewrite step-size: $\boldsymbol{\delta}_p = -\left(H_M(\boldsymbol{p}_0)\right)^{-1} \cdot \nabla_{\boldsymbol{p}} M(\boldsymbol{p}_0)$

as: $$\boldsymbol{\delta}_p = -\left(\sum_{(x,y,t)\in f} X^T H_{C^{(x,y,t)}(\boldsymbol{u}_0)} X\right)^{-1} \cdot \sum_{(x,y,t)\in f} X^T \nabla_{\boldsymbol{u}} C^{(x,y,t)}(\boldsymbol{u}_0)$$

And then to get rid of outliers....

Alignment Algorithm

how much to warp!?

$$\boldsymbol{\delta}_p = -\left(\sum_{(x,y,t)\in S} w(\boldsymbol{u}_0) X^T H_{C(\boldsymbol{u}_0)} X \right)^{-1} \cdot \sum_{(x,y,t)\in S} w(\boldsymbol{u}_0) X^T \nabla_{\boldsymbol{u}} C(\boldsymbol{u}_0)$$

$$w(\boldsymbol{u}_0) = w^{(x,y,t)}(\boldsymbol{u}_0) = -\left|H_{C(\boldsymbol{u}_0)}\right|.$$

Confidence-Weighted Regression

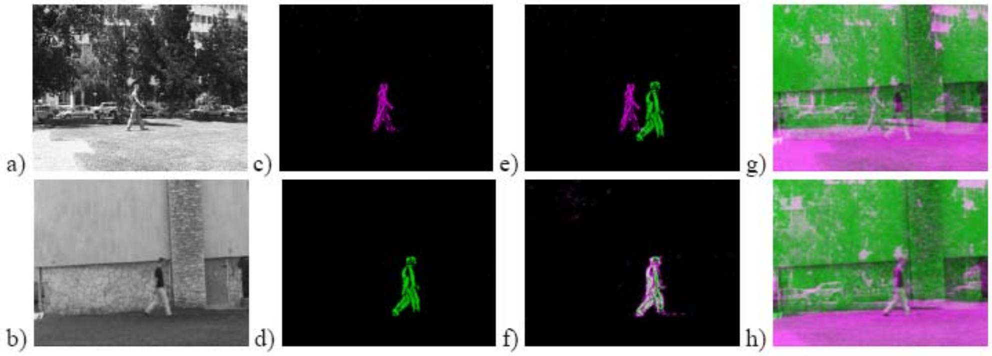
Give points more weight if they result in higher correlation!

$$H_{C^{(x,y,t)}} = \begin{bmatrix} \frac{\partial^2 C^{(x,y,t)}}{\partial x^2} & \frac{\partial^2 C^{(x,y,t)}}{\partial x \partial y} & \frac{\partial^2 C^{(x,y,t)}}{\partial x \partial t} \\ \frac{\partial^2 C^{(x,y,t)}}{\partial y \partial x} & \frac{\partial^2 C^{(x,y,t)}}{\partial y^2} & \frac{\partial^2 C^{(x,y,t)}}{\partial y \partial t} \\ \frac{\partial^2 C^{(x,y,t)}}{\partial t \partial x} & \frac{\partial^2 C^{(x,y,t)}}{\partial t \partial y} & \frac{\partial^2 C^{(x,y,t)}}{\partial t^2} \end{bmatrix}$$

Applications & Results

f:

g:



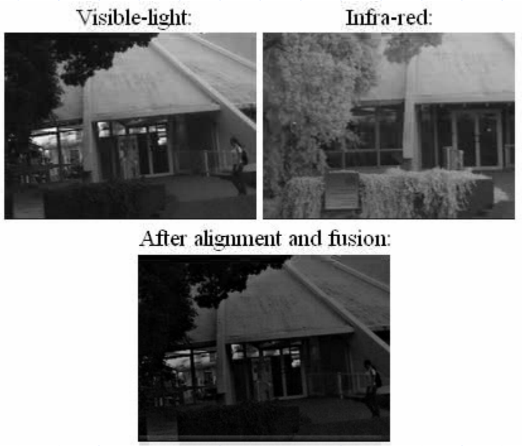
1. Backgrounds are different
2. Spatial scale of the walking person is different (by 36%)
3. Walking speed is different (by 13%)
4. Clothing colors are different

Applications & Results

Multi-Sensor Alignment

Common information is **details** in the scene (high frequency information both in **time** and **space**)

(photometric information is different → direction important!)



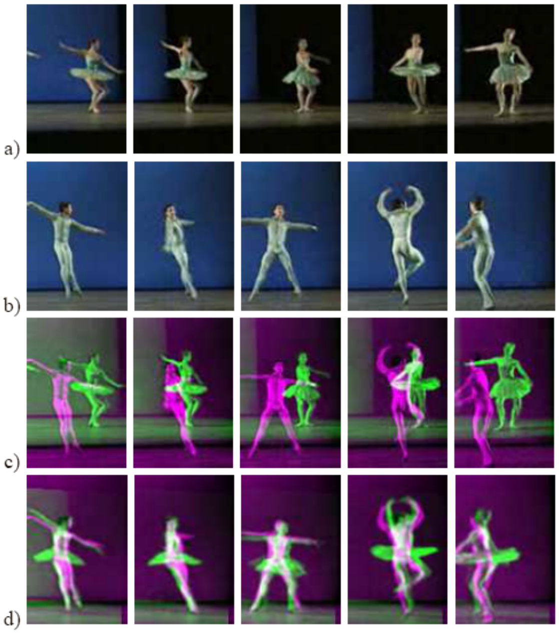
$$M(f,g) = M(f_x^{abs}, g_x^{abs}) + M(f_y^{abs}, g_y^{abs}) + M(f_t^{abs}, g_t^{abs})$$

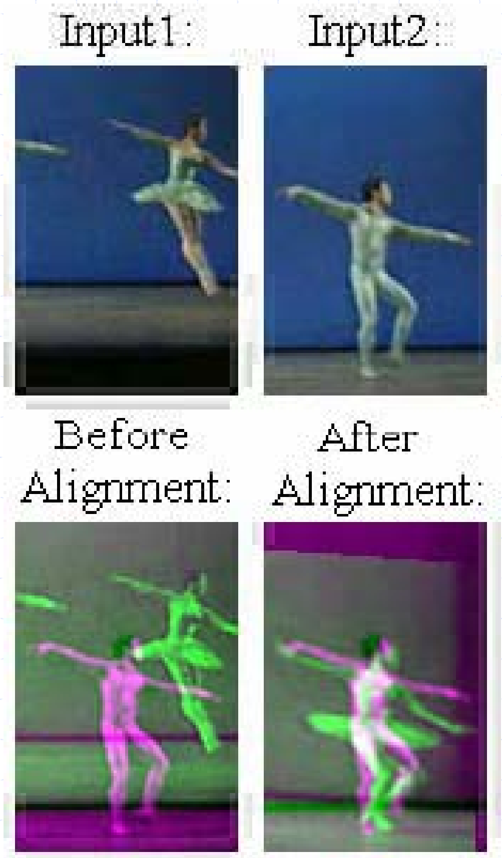
Applications & Results

Action Alignment

Common information is temporal variations.

$$M(f,g) = M\left(f_t^{abs}, g_t^{abs}\right)$$





Applications & Results

Applications

- Action/Event recognition
- Identification of people by behavior
- Comparing performance and style of people in sports!

Applications & Results

Action Alignment vs. Background Alignment

Only actions are aligned,
backgrounds are not aligned.

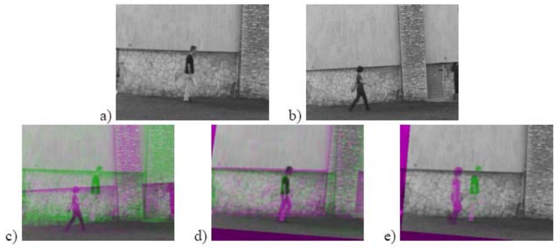
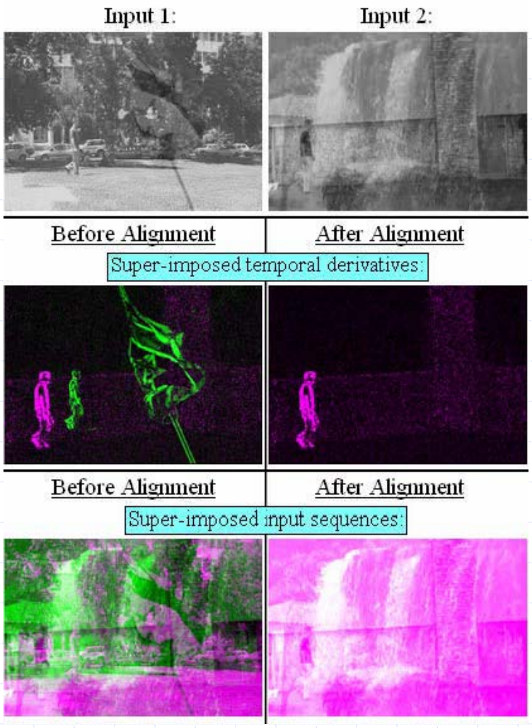


Fig. 3. Action alignment vs. background alignment. (a) and (b) show frame 45 of the two input sequences. (c) Initial misalignment (superposition of (a) and (b)). (d) Superposition after space-time alignment using temporal derivatives only (Eq. (10)). (e) Superposition after space-time alignment using spatial derivatives only (Eq. (11)). **For color figure and full video sequence see http://www.wisdom.weizmann.ac.il/∼vision/SpaceTimeCorrelations.html.**

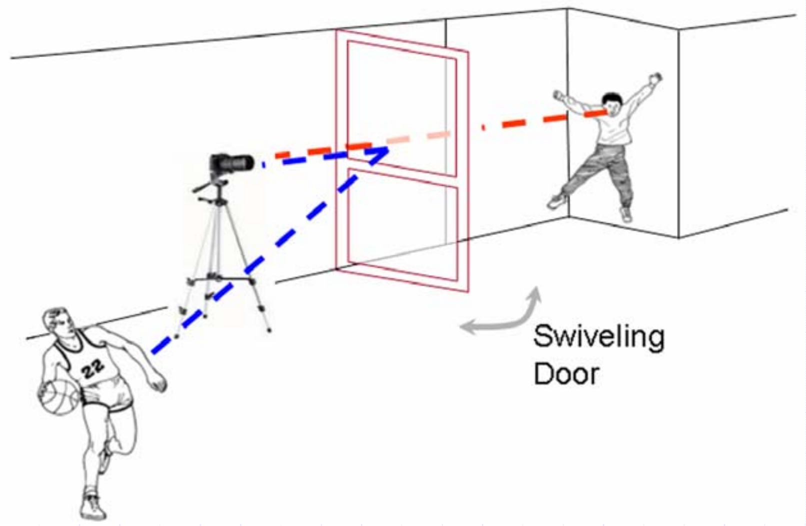
$$M(f,g) = M\left(f_x^{abs}, g_x^{abs}\right) + M\left(f_y^{abs}, g_y^{abs}\right)$$

Robustness & Locking Property

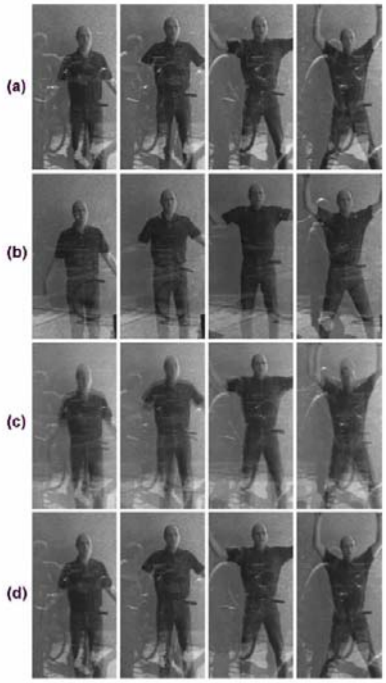


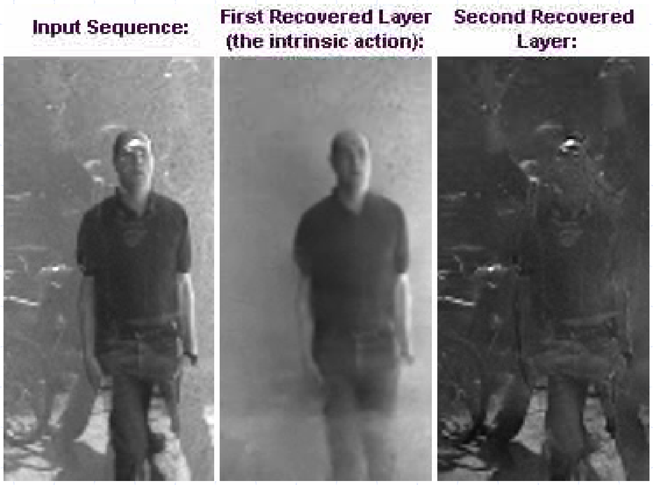
The outlier rejection part of the algorithm provides a strong **locking** property onto a dominant parametric motion!

Separating Transparent Layers in Real Video Transparency

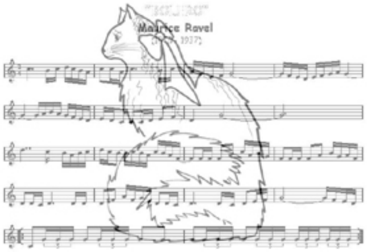
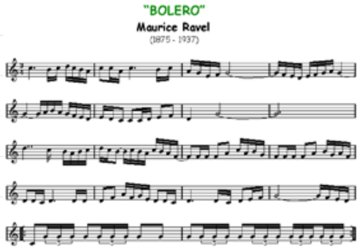


Separating Transparent Layers in Real Video Transparency





Aligning Audio & Separating Layers!

The Input Audio “Bolero” + Recited Poem about a Cat	The First Recovered Audio Track “Bolero”	The Second Recovered Audio Track Recited Poem about a Cat
		

References

•**M. Irani and P. Anandan.** *Robust Multi-Sensor Image Alignment*. In *IEEE International Conference on Computer Vision* (*ICCV*). India, January 1998

•**B. Sarel and M. Irani.** *Separating Transparent Layers through Layer Information Exchange*. In *European Conference on Computer Vision* (*ECCV*). May 2004.

•**P. Viola and W. Wells III.** *Alignment by Maximization of mutual information*. In *International Conference on Computer Vision*. Pages 16-23. Cambridge, MA. June 1995.

Merci Beaucoup!

Muchas Gracias!

Thank you!!!

Gracie Mille!