Region-of-Interest Based Adaptation of Video to Mobile Devices

Tamir Nuriel and Prof. David Malah

Electrical Engineering Department
Technion - Israel Institute of Technology

March 3, 2010
System Goals

- Adjustment of video in Standard Definition (SD) resolution (e.g., 720x576) to the smaller resolution used in cellular phones or other mobile devices (CIF - 352x288 /QCIF - 176x144).

- The naive solution is to scale each frame to the desired size. Instead we should display only the Region Of Interest (ROI).

- Motivation - Objects must be at a sufficient size, so that they are easy to recognize.

- Improve user-perceived quality.

- We focus on news broadcasting and interview scenes.

- Developing an editor guided algorithm for ROI detection and tracking.
System Goals

- Adjustment of video in Standard Definition (SD) resolution (e.g., 720x576) to the smaller resolution used in cellular phones or other mobile devices (CIF - 352x288 /QCIF - 176x144).
- The naive solution is to scale each frame to the desired size.
System Goals

- Adjustment of video in Standard Definition (SD) resolution (e.g., 720x576) to the smaller resolution used in cellular phones or other mobile devices (CIF - 352x288 /QCIF - 176x144).
- The naive solution is to scale each frame to the desired size.
- Instead we should display only the Region Of Interest (ROI).
System Goals

- Adjustment of video in Standard Definition (SD) resolution (e.g., 720x576) to the smaller resolution used in cellular phones or other mobile devices (CIF - 352x288 /QCIF - 176x144).
- The naive solution is to scale each frame to the desired size.
- Instead we should display only the Region Of Interest (ROI).
- Motivation - Objects must be at a sufficient size, so that they are easy to recognize.
System Goals

- Adjustment of video in Standard Definition (SD) resolution (e.g., 720x576) to the smaller resolution used in cellular phones or other mobile devices (CIF - 352x288 /QCIF - 176x144).
- The naive solution is to scale each frame to the desired size.
- Instead we should display only the Region Of Interest (ROI).
- Motivation - Objects must be at a sufficient size, so that they are easy to recognize.
- Improve user-perceived quality.
System Goals

- Adjustment of video in Standard Definition (SD) resolution (e.g., 720x576) to the smaller resolution used in cellular phones or other mobile devices (CIF - 352x288 /QCIF - 176x144).
- The naive solution is to scale each frame to the desired size.
- Instead we should display only the Region Of Interest (ROI).
- Motivation - Objects must be at a sufficient size, so that they are easy to recognize.
- Improve user-perceived quality.
- We focus on news broadcasting and interview scenes.
System Goals

- Adjustment of video in Standard Definition (SD) resolution (e.g., 720x576) to the smaller resolution used in cellular phones or other mobile devices (CIF - 352x288 /QCIF - 176x144).
- The naive solution is to scale each frame to the desired size.
- Instead we should display only the Region Of Interest (ROI).
- Motivation - Objects must be at a sufficient size, so that they are easy to recognize.
- Improve user-perceived quality.
- We focus on news broadcasting and interview scenes.
- Developing an editor guided algorithm for ROI detection and tracking.
Proposed Tracking Method

- Estimating the global motion of the whole frame and estimating the local motion inside the ROI.
- The method is based only on the horizontal and vertical projections.
- Estimating motion parameters in the transform domain.
- We use only 1D transforms, and thus reduce complexity.
- We limit ourselves to a rectangular ROI.
Motion Model

- Global motion estimation - motion caused by the camera.
Global motion estimation - motion caused by the camera.

Three-parameter model, for zoom, pan and tilt.

\[ f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y) \]
Motion Model

- Global motion estimation - motion caused by the camera.

- Three-parameter model, for zoom, pan and tilt.

\[ f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y) \]

- Local motion - motion inside the ROI.

- Two-parameter model for shift up or down and shift left or right.
Motion Model

- Global motion estimation - motion caused by the camera.
- Three-parameter model, for zoom, pan and tilt.

\[ f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y) \]

- Local motion - motion inside the ROI.
- Two-parameter model for shift up or down and shift left or right.
- First, updating the ROI according to the global motion parameters.
Global motion estimation - motion caused by the camera.

Three-parameter model, for zoom, pan and tilt.

\[ f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y) \]

Local motion - motion inside the ROI.

Two-parameter model for shift up or down and shift left or right.

First, updating the ROI according to the global motion parameters.

Next, updating the ROI according to the local motion parameters.
Scale Estimation using Cross-Correlation of Slices

- Transforming the model equation to the frequency domain.

\[
f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y)
\]

\[
F_2(f_x, f_y) = \frac{1}{\alpha^2} e^{j2\pi(f_x d_x + f_y d_y)} F_1 \left( \frac{f_x}{\alpha}, \frac{f_y}{\alpha} \right)
\]
Scale Estimation using Cross-Correlation of Slices

1. Transforming the model equation to the frequency domain.

\[ f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y) \]

\[ F_2(f_x, f_y) = \frac{1}{\alpha^2} e^{j2\pi(f_x d_x + f_y d_y)} F_1\left(\frac{f_x}{\alpha}, \frac{f_y}{\alpha}\right) \]

2. Taking the magnitudes leaves only the scale information.
Scale Estimation using Cross-Correlation of Slices

- Transforming the model equation to the frequency domain.

\[ f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y) \]

\[ F_2(f_x, f_y) = \frac{1}{\alpha^2} e^{j2\pi(f_x d_x + f_y d_y)} F_1 \left( \frac{f_x}{\alpha}, \frac{f_y}{\alpha} \right) \]

- Taking the magnitudes leaves only the scale information.

- Definition for slice \((f_y = 0)\):

\[ S_2(f_x) = F_2(f_x, 0). \]
Transforming the model equation to the frequency domain.

\[ f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y) \]

\[ F_2(f_x, f_y) = \frac{1}{\alpha^2} e^{j2\pi(f_x d_x + f_y d_y)} F_1 \left( \frac{f_x}{\alpha}, \frac{f_y}{\alpha} \right) \]

Taking the magnitudes leaves only the scale information.

Definition for slice \((f_y = 0)\): \(S_2(f_x) = F_2(f_x, 0)\).

The relation between slices:

\[ |S_2(f_x)| = |F_2(f_x, 0)| = \frac{1}{\alpha^2} \left| F_1 \left( \frac{f_x}{\alpha}, 0 \right) \right| = \frac{1}{\alpha^2} \left| S_1 \left( \frac{f_x}{\alpha} \right) \right| \]
Scale Estimation using Cross-Correlation of Slices

- Transforming the model equation to the frequency domain.

\[ f_2(x, y) = f_1(\alpha x + d_x, \alpha y + d_y) \]
\[ F_2(f_x, f_y) = \frac{1}{\alpha^2} e^{i2\pi(f_x d_x + f_y d_y)} F_1 \left( \frac{f_x}{\alpha}, \frac{f_y}{\alpha} \right) \]

- Taking the magnitudes leaves only the scale information.

- Definition for slice \((f_y = 0)\):
  \[ S_2(f_x) = F_2(f_x, 0). \]

- The relation between slices:

\[ |S_2(f_x)| = |F_2(f_x, 0)| = \frac{1}{\alpha^2} \left| F_1 \left( \frac{f_x}{\alpha}, 0 \right) \right| = \frac{1}{\alpha^2} \left| S_1 \left( \frac{f_x}{\alpha} \right) \right| \]

- Scale is estimated by using cross-correlation of slices in the frequency domain.

\[ |S_2(log(f_x))| = \frac{1}{\alpha^2} |S_1(log(f_x) - log(\alpha))| \]
Projection-Slice theorem

Projection (on the x-axis):

\[ P(x) = \int_{-\infty}^{\infty} f(x, y) \, dy \]
Projection-Slice theorem

- Projection (on the x-axis):
  \[ P(x) = \int_{-\infty}^{\infty} f(x, y) dy \]

- A slice through the origin of the 2D Fourier transform of a function, is the 1D Fourier transform of the projection, which is parallel to the slice.
Projection-Slice theorem

- Projection (on the x-axis):
  \[ P(x) = \int_{-\infty}^{\infty} f(x, y) \, dy \]

- A slice through the origin of the 2D Fourier transform of a function, is the 1D Fourier transform of the projection, which is parallel to the slice.

- No need to calculate the entire 2D Fourier transform.
Projection-Slice theorem

- Projection (on the x-axis):
  \[ P(x) = \int_{-\infty}^{\infty} f(x, y) dy \]

- A slice through the origin of the 2D Fourier transform of a function, is the 1D Fourier transform of the projection, which is parallel to the slice.
- No need to calculate the entire 2D Fourier transform.
- Instead, project on the perpendicular axis and take the Fourier transform.

![Diagram](projection-slice-theorem.png)
Scale Estimation Example

Original Frame

Frame after scale change of 0.9

Projections

Fourier transform magnitude of projections

Normalized frequency [radians] - logarithmic (base 10) scale

Scale Estimation Example
For sufficient scale resolution, we need many points in the cross-correlation.
Scale Estimation using Frequency-domain Mellin Transform

- For sufficient scale resolution, we need many points in the cross-correlation.
- Definition - One-dimensional Mellin transform:

\[ M(s) = \int f(x)x^{s-1}dx \]
Scale Estimation using Frequency-domain Mellin Transform

- For sufficient scale resolution, we need many points in the cross-correlation.
- Definition - One-dimensional Mellin transform:
  \[ M(s) = \int f(x)x^{s-1}dx \]

- The relation between slices
  \[ |S_2(f_x)| = \frac{1}{\alpha^2} \left| S_1 \left( \frac{f_x}{\alpha} \right) \right| \]
Scale Estimation using Frequency-domain Mellin Transform

- For sufficient scale resolution, we need many points in the cross-correlation.
- Definition - One-dimensional Mellin transform:
  \[ M(s) = \int f(x)x^{s-1}dx \]

- The relation between slices
  \[ |S_2(f_x)| = \frac{1}{\alpha^2} \left| S_1 \left( \frac{f_x}{\alpha} \right) \right| \]

- The relation between the Mellin transform of the absolute values of the slices
  \[ M_2(s) = \int |S_2(x)|x^{s-1}dx = \frac{1}{\alpha^2} \int \left| S_1 \left( \frac{x}{\alpha} \right) \right| x^{s-1}dx = \alpha^{s-2} M_1(s) \]
Scale Estimation using Frequency-domain Mellin Transform

- For sufficient scale resolution, we need many points in the cross-correlation.
- Definition - One-dimensional Mellin transform:
  \[ M(s) = \int f(x)x^{s-1}dx \]

- The relation between slices
  \[ |S_2(f_x)| = \frac{1}{\alpha^2} \left| S_1 \left( \frac{f_x}{\alpha} \right) \right| \]

- The relation between the Mellin transform of the absolute values of the slices
  \[ M_2(s) = \int |S_2(x)|x^{s-1}dx = \frac{1}{\alpha^2} \int \left| S_1 \left( \frac{x}{\alpha} \right) \right| x^{s-1}dx = \alpha^{s-2}M_1(s) \]

- Estimate the scale using a specific value of \( s \) (\( s \neq 2 \)):
  \[ \alpha = \left[ \frac{M_1(s)}{M_2(s)} \right]^{\frac{1}{2-s}} \]
The global motion causes some part of the frame not to appear in the next frame.
Effect of Fixed-Size Display

- The global motion causes some part of the frame not to appear in the next frame.
- Example for zoom in. The projection without the display limitation is different than the projection with the limitation.

![Fixed-size Display Effects on the Projections](image-url)
Reducing Effect of Fixed-Size Display

- We can solve this problem by zeroing out the parts of the each frame that do not appear in the other frame.
Reducing Effect of Fixed-Size Display

- We can solve this problem by zeroing out the parts of the each frame that do not appear in the other frame.

- In practice we don’t have the true parameters. Hence, we use the estimated scale and shifts from previous frame.
Reducing Effect of Fixed-Size Display

- We can solve this problem by zeroing out the parts of the each frame that do not appear in the other frame.
- In practice we don’t have the true parameters. Hence, we use the estimated scale and shifts from previous frame.
- We assume they are good initial estimates for the current parameters.
Reducing Effect of Fixed-Size Display

- We can solve this problem by zeroing out the parts of the each frame that do not appear in the other frame.
- In practice we don’t have the true parameters. Hence, we use the estimated scale and shifts from previous frame.
- We assume they are good initial estimates for the current parameters.

![Image (a)](image-a.png) is the previous frame.

![Image (b)](image-b.png) is the current frame, which is a scaled and shifted version of the previous frame.

![Image (c)](image-c.png) is the previous frame after zeroing out the parts that do not appear in the current frame.

![Image (d)](image-d.png) is the current frame after zeroing out the parts that do not appear in the previous frame.
Scale Estimation using Spatial-Domain Mellin transform

- Perform projections after reducing fixed-size effects.
Scale Estimation using Spatial-Domain Mellin transform

- Perform projections after reducing fixed-size effects.
- Good estimates of the pan and tilt allow us to avoid transforming to the frequency domain.
Perform projections after reducing fixed-size effects.

Good estimates of the pan and tilt allow us to avoid transforming to the frequency domain.

The relation between the projections, after shift correction.

\[ P_2(x) = P_1(\alpha x) \]
Perform projections after reducing fixed-size effects.

Good estimates of the pan and tilt allow us to avoid transforming to the frequency domain.

The relation between the projections, after shift correction.

\[ P_2(x) = P_1(\alpha x) \]

The relation between the Mellin transforms of the projections

\[ M_{p_2}(s) = \int P_2(x)x^{s-1} \, dx = \int P_1(\alpha x)x^{s-1} \, dx = \alpha^{-s} M_{p_1}(s) \]
Scale Estimation using Spatial-Domain Mellin transform

- Perform projections after reducing fixed-size effects.
- Good estimates of the pan and tilt allow us to avoid transforming to the frequency domain.
- The relation between the projections, after shift correction.

\[ P_2(x) = P_1(\alpha x) \]

- The relation between the Mellin transforms of the projections

\[ M_{p_2}(s) = \int P_2(x)x^{s-1} \, dx = \int P_1(\alpha x)x^{s-1} \, dx = \alpha^{-s} M_{p_1}(s) \]

- Therefore, the scale estimation is

\[ \alpha = \left[ \frac{M_{p_1}(s)}{M_{p_2}(s)} \right]^{\frac{1}{s}} \]
Perform projections after reducing fixed-size effects.

Good estimates of the pan and tilt allow us to avoid transforming to the frequency domain.

The relation between the projections, after shift correction:

\[ P_2(x) = P_1(\alpha x) \]

The relation between the Mellin transforms of the projections:

\[ M_{p_2}(s) = \int P_2(x)x^{s-1} \, dx = \int P_1(\alpha x)x^{s-1} \, dx = \alpha^{-s} M_{p_1}(s) \]

Therefore, the scale estimation is

\[ \alpha = \left( \frac{M_{p_1}(s)}{M_{p_2}(s)} \right)^{\frac{1}{s}} \]

Shift values for current frame will be estimated after the scale estimation.
Global Translation Estimation

- After estimating the scale factor, we can estimate the horizontal and vertical shifts.
Global Translation Estimation

- After estimating the scale factor, we can estimate the horizontal and vertical shifts.
- First, scale the projection from the previous frame with the estimated scale.
Global Translation Estimation

- After estimating the scale factor, we can estimate the horizontal and vertical shifts.
- First, scale the projection from the previous frame with the estimated scale.
- Then perform cross-correlation between the scaled projection and the projection from the current frame.
Global Translation Estimation

- After estimating the scale factor, we can estimate the horizontal and vertical shifts.
- First, scale the projection from the previous frame with the estimated scale.
- Then perform cross-correlation between the scaled projection and the projection from the current frame.
- The global motion estimation can be done iteratively, but we didn’t find it necessary.
The ROI window is updated according to the estimated global motion parameters.
The ROI window is updated according to the estimated global motion parameters.

We assume that there is no local scale change.
Local Translation Estimation

- The ROI window is updated according to the estimated global motion parameters.
- We assume that there is no local scale change.
- Calculate horizontal and vertical projections only for the ROI.
The ROI window is updated according to the estimated global motion parameters.

We assume that there is no local scale change.

Calculate horizontal and vertical projections only for the ROI.

Scale the projection from the previous ROI and then perform cross-correlation in the spatial-domain.
Example - TV scenes
We developed a novel algorithm for tracking the ROI based on projections. We used a pan, tilt and zoom camera model and estimate its parameters in the frequency domain.
We developed a novel algorithm for tracking the ROI based on projections. We used a pan, tilt and zoom camera model and estimate its parameters in the frequency domain.

We reduced complexity by using the slice-projection theorem and the Mellin transform.
Summary

- We developed a novel algorithm for tracking the ROI based on projections. We used a pan, tilt and zoom camera model and estimate its parameters in the frequency domain.
- We reduced complexity by using the slice-projection theorem and the Mellin transform.
- We further reduced complexity by using Mellin transform in the spatial domain.
Thank You!