# Multimodal Computational Methods in Political Science

## Motion Estimation in Video Analysis

**Clara Fochler**

Ludwig-Maximilians-Universität München

June 02, 2026

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Agenda

Today: Video Analysis

From Images to Videos

Motion Estimation

Motion Detection in Political Science Research - An Example

Next Session

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Today: Video Analysis

- Video analysis is commonly used for object detection, pose estimation, activity recognition and person re-identification
- We analyze a series of images - we already know methods to analyse images - CNNs (classification) and R-CNNs/YOLO (object detection) As for images, for video analysis there are also some open-source autotaggers OpenPose (for pose estimation) and PyTorch offers multiple models for e.g. instance segmentation, and action recognition

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# From Images to Videos

- Videos are sequences of frames –> processing them demands capturing motion/temporal information
- Standard image-based transformers only handle spatial information per frame, ignoring time
- Temporal relationships are critical for true video understanding –> requires the additional axis: **time** in the input representation
- Today we will cover motion - we will focus on the multi-modal characteristic in our last session

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Motion Estimation: Error Metric and Search Strategy

- Motion estimation aims to determine the displacement between two or more images
- This requires:
  - selecting an appropriate **error (similarity) metric** to compare images, and
  - employing a **search strategy** to identify the motion parameters that minimize the error

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Translational Alignment (2D shift)

Let us assume the parts of first image are in the second image as well:

- Image alignment can be formulated as finding the relative shift between two images or image patches.
- Estimates the displacement between a template and a target image using a translation vector $\mathbf{u} = (u, v)$.
- Assumes **brightness constancy**: corresponding pixels are (approximately) preserved across images
- Displacement $\mathbf{u}$ may be fractional, requiring interpolation of $I_1(\mathbf{x})$ (e.g., bilinear or bicubic)
- For colour images, differences are typically aggregated over channels or computed in alternative colour spaces; alternatively, only luminance may be used

$$e_i = I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i) \quad \text{(residual error/displaced frame difference)}$$

$$E_{\mathrm{SSD}}(\mathbf{u}) = \sum_i e_i^2 = \sum_i [I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Spatially Varying Weights in Sum of Squared Differences (SSD) I

- Standard SSD assumes all pixels contribute equally and ignores image boundaries
- In practice, pixels may be invalid (outside boundaries) or intentionally downweighted (e.g., foreground removal, background stabilization)
- This can be achieved by assigning spatially varying per-pixel weights to both images, resulting in a weighted (windowed) SSD formulation

$$E_{\mathrm{WSSD}}(\mathbf{u}) = \sum_i w_0(\mathbf{x}_i)\ w_1(\mathbf{x}_i + \mathbf{u})\ [I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

- With large motion search ranges, SSD may favor solutions with small overlap $\rightarrow$ reduced by normalizing with the overlap area $A$.

# Spatially Varying Weights in Sum of Squared Differences (SSD) II

$$A = \sum_i w_0(\mathbf{x}_i) w_1(\mathbf{x}_i + \mathbf{u})$$

- The final score can be expressed as a per-pixel/mean squared error by normalizing with the overlap area $A$

$$\mathrm{RMSE} = \sqrt{\frac{E_{\mathrm{WSSD}}(\mathbf{u})}{A}}$$

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Bias and Gain (Exposure Differences)

- Images may differ in exposure, causing global intensity changes between frames
- This is modelled using an affine intensity transformation (bias and gain):

$$I_1(\mathbf{x} + \mathbf{u}) = (1 + \alpha)\, I_0(\mathbf{x}) + \beta$$

- $\alpha$: gain (contrast scaling), $\beta$: bias (intensity offset)
- Alignment is then formulated as a least-squares regression problem:

$$E_{\mathrm{BG}}(\mathbf{u}) = \sum_i \Big[ I_1(\mathbf{x}_i + \mathbf{u}) - (1 + \alpha) I_0(\mathbf{x}_i) - \beta \Big]^2 = \sum_i \Big[ \alpha I_0(\mathbf{x}_i) + \beta - e_i \Big]^2$$

- Equivalent to estimating motion + linear intensity correction jointly

# Correlation

- Instead of measuring alignment via intensity differences, one can use **correlation-based matching**
- The basic idea is to maximize the overlap between two image patches:

$$ECC(u) = \sum_i I_0(x_i)\, I_1(x_i + u)$$

- However, plain cross-correlation is sensitive to brightness variations:
  - very bright regions can dominate the matching score
  - bias and gain differences are not properly handled
- A more robust alternative is **normalized cross-correlation (NCC)**:

$$ENCC(u) = \frac{\sum_i (I_0(x_i) - \overline{I}_0)(I_1(x_i + u) - \overline{I}_1)}{\sqrt{\sum_i (I_0(x_i) - \overline{I}_0)^2}\sqrt{\sum_i (I_1(x_i + u) - \overline{I}_1)^2}}$$

- NCC is invariant to linear intensity changes and is bounded in $[-1, 1]$
- A related variant is normalized SSD, which yields similar performance but can be computationally more efficient in some settings

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Structural Similarity Index (SSIM)

Instead of finding the best alignment (like SSD/NCC) → quantifying similarity between two frames (formula taken from Dietrich 2021):

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{1}$$

- $\mu_x, \mu_y$: means of pixel matrices → **luminance comparison** (cf. Bias/Gain, Slide 9)
- $\sigma_x, \sigma_y$: standard deviations → **contrast comparison**
- $\sigma_{xy}$: cross-correlation → **structural comparison** (NCC, Slide 10)
- $C_1, C_2$: small stabilizing constants (prevents division by zero)
- $SSIM \in [-1, 1]$, where 1 = identical frames

We will see how this was used in Political Science research at the end of the seminar.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Hierarchical, Fourier, and Incremental Alignment

How do we efficiently find the best alignment between two images (i.e. minimize matching error)?

## Hierarchical Search

- build image pyramid (low $\rightarrow$ high resolution)
- solve alignment at coarse scale first
- refine result at finer levels
- reduces search space dramatically
- robust to large motions

## Fourier-based Alignment

- alignment in frequency domain
- shifts become phase changes
- use FFT for efficient computation
- avoids exhaustive spatial search
- useful for large global translations

## Incremental Refinement

- starts from coarse alignment
- improves precision step-by-step
- uses local image gradients
- small updates ($\Delta u$)
- reaches sub-pixel accuracy

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Hierarchical Motion Estimation

- Build an image pyramid (low to high resolution)
- Estimate motion on coarse scale first
- Refine step-by-step on finer levels
- Each level only searches a small neighborhood

## Coarse-to-fine update

$$\hat{u}^{(l-1)} = 2u^{(l)} \tag{2}$$

- $u^{(l)}$: motion estimate on coarse level (low-resolution image)
- $\hat{u}^{(l-1)}$: initial estimate for the next finer level
- factor 2: scaling due to doubled spatial resolution between pyramid levels

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Fourier-based alignment

When the required shift is large (e.g. image stitching), coarse-to-fine pyramids may fail due to loss of detail. Instead, alignment can be solved efficiently in the Fourier domain.

- **Shift becomes phase change**

$$I_1(x+u) \iff I_1(\omega)\, e^{-ju\cdot\omega}$$

  A spatial shift becomes a linear phase change in frequency space.

- **Correlation via Fourier transform**

$$ECC(u) = \mathcal{F}^{-1}\{I_0(\omega)\, I_1^*(\omega)\}$$

  This replaces expensive spatial search with FFT-based computation.

- **Robust variants**
  - Windowed correlation: handles partial overlap
  - Phase correlation: normalizes magnitude, uses phase only $\rightarrow$ sharp peak at correct shift

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Incremental Refinement (Sub-pixel Alignment)

Pixel-level search is often not sufficient for image stabilization or stitching. Incremental refinement improves accuracy by iteratively updating small displacement corrections.

- We want sub-pixel alignment
  - Refine an initial estimate $u$ to higher precision
  - Used after coarse search or pyramid methods
- **Local linearization (Taylor expansion)**

$$I_1(x_i + u + \Delta u) \approx I_1(x_i + u) + J_1(x_i + u)\,\Delta u$$

$$e_i = I_1(x_i + u) - I_0(x_i)$$

- **Least-squares update (Lucas–Kanade)**

$$A\Delta u = b$$

where

$$A = \sum_i J_1^T J_1, \qquad b = -\sum_i e_i J_1$$

  - Iterative gradient descent $\rightarrow$ improves sub-pixel accuracy
  - Works best near correct alignment
  - Sensitive to texture (aperture problem $\rightarrow$ ill-conditioned $A$)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Parametric Motion I

Most videos we analyze today come from:

- smartphones (handheld, shaky recordings)
- social media platforms (TikTok, Instagram, YouTube)
- surveillance or crowd-sourced footage

These videos do not only shift left/right:

- camera rotates slightly
- zoom changes (walking, moving closer/farther)
- perspective effects appear (not just translation)

# Parametric Motion II

–> extend translational alignment
Instead of a single displacement $u$, motion is described by a parameter vector $p$:

$$x'(x; p)$$

This can represent:

- translation
- affine motion (rotation, scaling, shear)
- homography (perspective effects)

# Parametric Motion III

**Alignment objective (SSD error)**
We still compare image intensities under the motion model:

$$E_{PM}(p) = \sum_i \left[ I_1(x'(x_i; p)) - I_0(x_i) \right]^2$$

**Key difficulty:** this is non-linear in $p$

**Linearization (Taylor expansion)**
We approximate around current parameters $p$:

$$I_1(x'(x_i; p + \Delta p)) \approx I_1(x_i') + J_1(x_i')\Delta p$$

which gives:

$$\sum_i \left[ J_1(x_i')\Delta p + e_i \right]^2$$

# Parametric Motion IV

where:

$$e_i = I_1(x_i') - I_0(x_i)$$

and the Jacobian is:

$$J_1(x_i') = \nabla I_1(x_i') \frac{\partial x'}{\partial p}$$

# Parametric Motion V

**Gauss–Newton update**

The linearized problem leads to:

$$A\Delta p = b$$

with:

$$A = \sum_i J^T J, \quad b = -\sum_i J^T e$$

**Iterative procedure:**

- compute residuals under current motion $p$
- solve for small correction $\Delta p$
- update $p \leftarrow p + \Delta p$
- repeat until convergence

# Optical Flow I

## What is Optical Flow?

- **Most general and challenging** form of motion estimation
- Compute an independent motion estimate for **each individual pixel** (dense motion estimation)
- Involves minimizing brightness or color differences between consecutive frames

$$E_{\text{OF}} = \sum_i [I_1(x_i + u_i) - I_0(x_i)]^2 \tag{3}$$

- **The Core Challenge (Underconstrained):** The number of unknown displacement variables ($u_i$) is twice the number of available measurements per pixel.

# Optical Flow II

## The Two Classic Solution Approaches

- **Local (Patch-Based) Methods:** Summation over overlapping regions/windows (e.g., Lucas-Kanade 1981, using Taylor series  coarse-to-fine pyramids)
- **Global Methods:** Adding spatial smoothness terms/regularization over the entire image to find a global minimum (e.g., Horn & Schunck 1981)

# Optical Flow III

## Local vs. Global  Advanced Extensions

- **Motion Discontinuities:** Global optimization usually performs better near motion boundaries because constraints aren't averaged over patches.
- **Robust Regularization:** $L_1$ norm (Total Variation) or anisotropic smoothness priors help preserve edges and yield convex energies.
- **Hybrid Models & Parametric Motion:** Combining segment/global models (like rigid camera motion) with per-pixel residuals.
- **Modern Paradigm:** Classic energy-minimization and coarse-to-fine approaches have largely been surpassed by **deep neural networks** (trained on datasets like MPI Sintel or KITTI).

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Deep Learning for Optical Flow I

- Deep neural networks are now essential for high-performance optical flow
- Early idea: DeepFlow uses convolution-like feature aggregation + classical variational optimization

## First end-to-end learning: FlowNet

- FlowNetS: encoder-decoder CNN trained on synthetic FlyingChairs data
- FlowNetC: uses correlation (cost volume) between features
- FlowNet 2.0: cascaded networks + image warping + refinement

# Deep Learning for Optical Flow II

## Coarse-to-fine learning approaches

- SPyNet: pyramid + warping + refinement across scales
- PWC-Net:
  - feature pyramids for both images
  - warping using previous flow estimate
  - cost volume via feature correlation
  - CNN-based refinement per level
  - final context network with dilated convolutions
- Strong correspondence between classical coarse-to-fine methods and deep learning architectures
- Deep models replace explicit energy minimization with learned refinement

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Layered Motion

- Motion often caused by a small number of objects at different depths
- For layer motion detection pixels are thus grouped into layers (objects / depth layers)
- Each layer has appearance + parametric motion
- Layers are alpha-matted (sprites / video objects)
- Scene is reconstructed by compositing layers (back-to-front)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Motion Detection in Political Science Research - An Example I

Dietrich, Bryce J. “Using Motion Detection to Measure Social Polarization in the U.S. House of Representatives.” Political Analysis 29, no. 2 (2021): 250–59. https://doi.org/10.1017/pan.2020.25.

- Manually identified 17,700 "good" frames from a random sample → video hashing (16 hexadecimal characters) to compare each frame with every other frame → utilized a computing cluster to calculate 113,935,802,380 pairwise comparisons → frames sharing at least 10 hexadecimal characters (hash values) with at least one of the "good" frames were classified as overhead shots.
- Used motion detection to operationalize the dependent variable (DV): created approximately 17 relevant clips for each video and then measured sequence differences using structural similarity (SSIM)
- High SSIM between consecutive frames = low motion = polarization

**Next session: Learning representations for video data**

- CNNs for spatial (and short-term spatio-temporal) feature extraction
- RNNs for temporal modeling
- Video Vision Transformers (spatio-temporal attention models)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# REFERENCES


- Nyhuis, D., Ringwald, T., Rittmann, O., Gschwend, T., Stiefelhagen, R. (2021). Automated video analysis for social science research. In U. Engel, A. Quan-Haase, S. X. Liu, L. Lyberg (Eds.), Handbook of Computational Social Science, Volume 2: Data science, statistical modelling, and machine learning methods (pp. 386–398). Routledge. https://doi.org/10.4324/9781003025245-26
- Szeliski, R. (2022). Computer Vision: Algorithms and Applications (2nd ed.). Springer. Chapter 9: Motion Estimation. https://doi.org/10.1007/978-3-030-34372-9

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Thanks for your attention!