CVID - Cours 1

Nicolas Boutry[1]

↪ nicolas.boutry@lrde.epita.fr

[1] Laboratoire de Recherche et Développement de l'EPITA (LRDE), France

Feb 2023

Outline

Outline

Summary I

- Course 1
    - Optical Flow (apparent motion)
    - Motion Estimation (pixel-wise / block-wise)
    - Motion Estimation (node-based / mesh-based)
    - Brief presentation of the other possible approaches
- Course 2
    - Predictive Coding
    - Motion Compensation (MC)
    - Scalability and granularity
    - Video Compression

Scalable video coding I

- Scalability refers to the capability of recovering physically meaningful image or video information from decoding only partial compressed bitstreams,
  - Quality scalability: finer to finer quantizations,
  - Spatial scalability: different spatial resolutions (Laplacian Pyramid, ...),
  - Temporal scalability (we can jump frames and add the missing ones progressively),
  - Frequency scalability (lower frequencies to higher frequencies),
  - Combination of basic schemes
  - Granularity (coarse vs fine ones)
- Object-based scalability (different resolutions for different objects)

Outline

Optical flow

2D motion vs. optical flow I

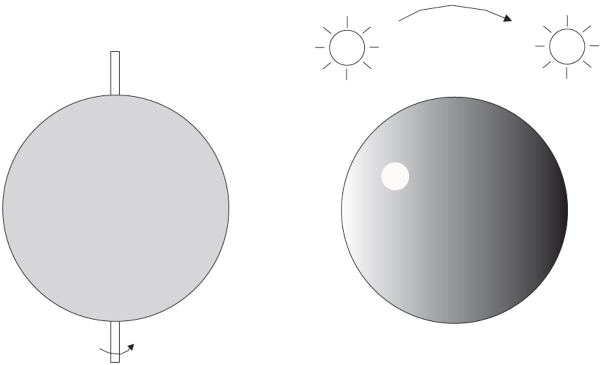


Figure 6.1. The optical flow is not always the same as the true motion field. In (a), a sphere is rotating under a constant ambient illumination, but the observed image does not change. In (b), a point light source is rotating around a stationary sphere, causing the highlight point on the sphere to rotate. Adapted from [17, Fig.12-2].

2D motion vs. optical flow II

- We can observe movements where there are not, and observe that there is no motion when there are!
- The observed or apparent 2D motion is referred to as optical flow in computer vision literature.
- In brief, the optical flow may not be the same as the true 2D motion.

Optical flow equation and ambiguity in motion estimation I

- Consider a video sequence whose luminance is variable in time $\psi(x, y, t)$.
- Assume that an point (x, y) at time t is moved to $(x + d_x, y + d_y)$ at time $t + d_t$.
- Under the constant intensity assumption, the images of the same object point at different times have the same luminance value:

$$\psi(x + d_x, y + d_y, t + d_t) = \psi(x, y, t)$$

- Using Taylor's expansion, when d_x, d_y and d_t are small, then we have:

$$\psi(x + d_x, y + d_y, t + d_t) = \psi(x, y, t) + \frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial t} d_t$$

- We obtain that:

$$\frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial t} d_t = 0,$$

which shows the relation between the motion vector (d_x, d_y) and d_t.

Optical flow equation and ambiguity in motion estimation II

- Let us define $v_x = d_x/d_t$ and $v_y = d_y/d_t$, then $\boldsymbol{v} := (v_x, v_y)$ is the velocity vector and we obtain that:

$$\frac{\partial \psi}{\partial x} \, v_x + \frac{\partial \psi}{\partial y} \, v_y + \frac{\partial \psi}{\partial t} = 0,$$

  which can be written:

$$\nabla \psi^T \, \boldsymbol{v} + \frac{\partial \psi}{\partial t} = 0,$$

  with $\nabla \psi = \left[\frac{\partial \psi}{\partial x}, \frac{\partial \psi}{\partial y} \right]^T$ is the spatial gradient vector of $\psi(x, y, t)$.

- From now on, we will write $\boldsymbol{x} := (x, y)$,

Optical flow equation and ambiguity in motion estimation III

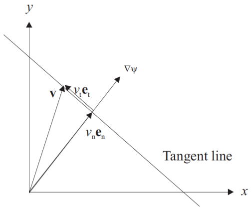


Figure 6.2. Decomposition of motion $\mathbf{v}$ into normal ($v_n\mathbf{e}_n$) and tangent $v_t\mathbf{e}_t$ components. Given $\boldsymbol{\nabla}\psi$ and $\frac{\partial\psi}{\partial t}$, any MV on the tangent line satisfies the optical flow equation.

- The flow vector $\boldsymbol{v}$ at any point x can be decomposed into two orthogonal components:

$$\boldsymbol{v} = v_n\ e_n + v_t\ e_t,$$

- As we can observe, when a straight edge moves in the plane, we can only detect the normal v_n of its motion vector !

Optical flow equation and ambiguity in motion estimation IV

- Because $\nabla\psi = ||\nabla\psi||\, e_n$, the optical flow equation can be rewritten as:

$$v_n\, ||\nabla\psi|| + \frac{\partial\psi}{\partial t} = 0,$$

where $||\nabla\psi||$ is the magnitude of the gradient vector.

- The consequences of these equations are:

1 (A) At any pixel x, one cannot determine the motion vector v based on $\nabla\psi$ and $\frac{\partial\psi}{\partial t}$ alone: <u>there is only one equation for two unknowns</u> (v_x and v_y, or v_n and v_t).

1 (B) In fact, the undetermined component is v_t.To solve both unknowns, one needs to impose additional constraints.

1 (C) The most common constraint is that the flow vectors should vary smoothly spatially (regularity).

2 (A) We can compute:

$$v_n = -\frac{\frac{\partial\psi}{\partial t}}{||\nabla\psi||},$$

whatever the value of v_t.

Optical flow equation and ambiguity in motion estimation V

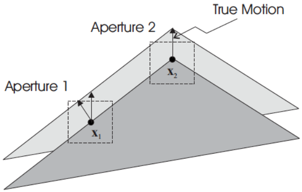


Figure 6.3. The aperture problem in motion estimation: To estimate the motion at $\mathbf{x}_1$ using aperture 1, it is impossible to determine whether the motion is upward or perpendicular to the edge, because there is only one spatial gradient direction in this aperture. On the other hand, the motion at $\mathbf{x}_2$ can be determined accurately, because the image has gradient in two different directions in aperture 2. Adapted from [39, Fig. 5.7].

2 (B) This ambiguity in estimating the motion vector is known as the aperture problem.

2 (C) The motion can be estimated uniquely only if the aperture contains at least two different gradient directions.

Optical flow equation and ambiguity in motion estimation VI

3 (A) In regions with constant brightness so that $||\nabla\psi|| = 0$, the flow vector is indeterminate.

3 (B) This is because there is no perceived brightness changes when the underlying surface has a flat pattern.

3 (C) The estimation of motion is reliable only in regions with brightness variation, i.e., regions with edges or non-flat textures.

General methodologies

General methodologies I

- We consider the ME between two given frames, $\psi(x, y, t_1)$ and $\psi(x, y, t_2)$.
- The MV at x between time t_1 and t_2 is defined as the displacement of this point from t_1 to t_2.
- We will call the frame at time t_1 the tracked/reference frame, and the frame at t_2 the anchor/current frame.
- Depending on the intended application, the anchor frame can be either before or after the tracked frame in time.

General methodologies II

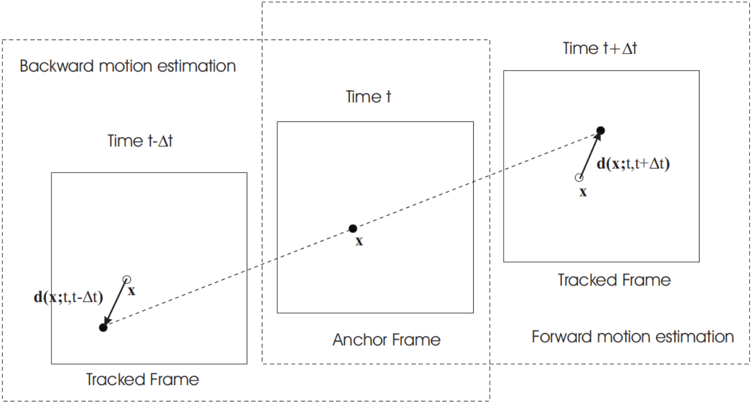


Figure 6.4. Forward and backward motion estimation. Adapted from [39, Fig. 5.5].

- The problem is referred as to as forward motion estimation when $t_1 < t_2$ and as backward motion estimation when $t_2 < t_1$.

General methodologies III

- For notation convenience, from now on, we use $\psi_1(\boldsymbol{x})$ and $\psi_2(\boldsymbol{x})$ to denote the reference and current frames respectively.
- In general, we can represent the MV as $d(x; \boldsymbol{a})$ where:

$$\boldsymbol{a} = [a_1, a_2, \ldots, a_n]^T$$

  is a vector containing all the motion parameters.
- Similarly, the mapping function can be denoted as $w(\boldsymbol{x}; \boldsymbol{a}) = \boldsymbol{x} + d(\boldsymbol{x}; \boldsymbol{a})$,
- The ME problem is to estimate the motion parameter vector $\boldsymbol{a}$.
- Methods that have been developed can be categorized into two groups: feature-based (matching) and intensity-based.
- We will only see the intensity-based approach.
- This approach applies the constant intensity assumption or the optical flow equation at every pixel.

Motion representation

Motion representation I

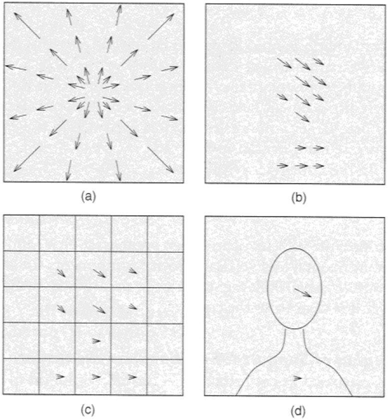


Figure 6.5. Different motion representations: (a) global, (b) pixel-based, (c) block-based, and (d) region-based. From [38, Fig. 3].

Motion representation II

- A key problem in ME is how to parameterize the motion field:
  - Translations,
  - Polynomial motions,
  - Rotations,
  - ...
- However, usually, there are multiple objects in the scene that move differently,
- When there are several objects in an image moving in different directions:
  - The most direct and unconstrained approach is to specify the motion vector at every pixel; this is the so called pixel-based representation.
  - It requires the estimation of a large numbers of unknowns (twice the number of pixels!)
  - The solution can often be physically incorrect!
  - Then we need physical constraints like regularity of the MV field.

Motion representation III

- If only the camera is moving or the imaged scene contains a single moving object within a planar surface:
  - One could use a global motion representation to characterize the entire motion field.
- In general:
  - It is more appropriate to divide an image frame into multiple regions so that the motion within each region can be characterized well by a parameterized model.
  - This is known as region-based motion representation
  - Very complicated in practice: do we estimate first the motion and then the regions? or the regions are then the motions? Or both in the same time?
  - To reduce the complexity: we can decompose the image into small blocks (block-based representation).

Motion representation IV

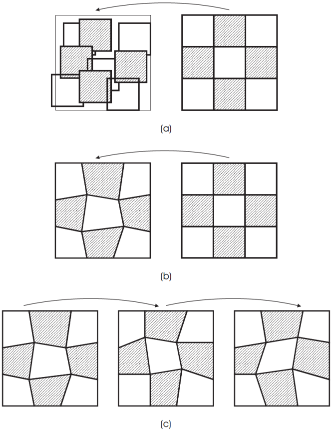


Figure 6.13. Comparison of block-based and mesh-based motion representations: (a) Block-based motion estimation between two frames, using a translational model within each block in the anchor frame; (b) Mesh-based motion estimation between two frames, using a regular mesh at the anchor frame; (d) Mesh-based motion tracking, using the tracked mesh for each new anchor frame.

Motion representation V

- The simplest version models the motion in each block by a constant translation (all over the block),
- The estimation problem becomes that of finding one MV for each block.
- This method provides a good compromise between accuracy and complexity.
- The block-based approach does not impose any constraint on the motion transition between adjacent blocks.
- The resulting motion is often discontinuous across block boundaries.
- One approach to overcome this problem is by using a mesh-based representation where we deform the image as a mesh in time.
- In this method, the image frame is partitioned into non-overlapping polygonal elements.
- This representation induces a motion field that is continuous everywhere.

Motion representation VI

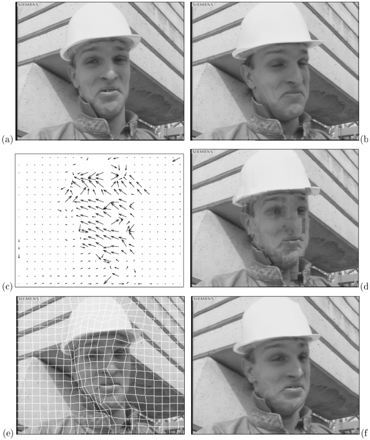


Figure 6.8. Example motion estimation results: (a) the tracked frame; (b) the anchor frame; (c-d) motion field and predicted image for the anchor frame (PSNR=29.86 dB) obtained by half-pel accuracy EBMA ; (e-f) motion field (represented by the deformed mesh overlaid on the tracked frame) and predicted image (PSNR=29.72 dB) obtained by the mesh-based motion estimation scheme in [43].

- It is not the ultimate solution: it can introduce warping effects!

Motion estimation criteria

Motion estimation criteria I

- We can simply minimize the error based on the Displaced Frame Difference (DFD):

$$E_{DFD}(a) = \sum_{\boldsymbol{x} \in \Lambda} \left| \psi_2(w(\boldsymbol{x}; \boldsymbol{a})) - \psi_1(\boldsymbol{x}) \right|^p,$$

where Λ is the domain of all pixels in ψ_1, and p is a positive number.

- When $p = 1$, the above error is called mean absolute difference (MAD), and when $p = 2$, the mean squared error (MSE).
- The error image $e(\boldsymbol{x}; \boldsymbol{a}) = \psi_2(w(\boldsymbol{x}; \boldsymbol{a})) - \psi_1(\boldsymbol{x})$ is usually called displaced frame difference (DFD) image.
- When a is optimal (case $p = 2$):

$$\frac{\partial E_{DFD}}{\partial \boldsymbol{a}} = 2 \sum_{\boldsymbol{x} \in \Lambda} (\psi_2(w(\boldsymbol{x}; \boldsymbol{a})) - \psi_1(\boldsymbol{x})) \frac{d(w(\boldsymbol{x}; \boldsymbol{a}))}{d\boldsymbol{a}} \nabla \psi_2(w(\boldsymbol{x}; \boldsymbol{a})) = 0,$$

Motion estimation criteria II

- Or we can minimize the error relative to the optical flow:
- Let $\psi_1(x,y) = \psi(x,y,t)$ and $\psi_2(x,y) = \psi(x,y,t+d_t)$. If d_t is small, we can assume that:

$$\frac{\partial \psi}{\partial t}\, d_t = \psi_2(x) - \psi_1(x),$$

- Then the optical flow equation becomes:

$$\frac{\partial \psi_1}{\partial x}\, d_x + \frac{\partial \psi_1}{\partial y}\, d_y + (\psi_2 - \psi_1) = 0,$$

or equivalently:

$$\nabla \psi_1^T d + (\psi_2 - \psi_1) = 0.$$

Motion estimation criteria III

- It is equivalent to minimize:

$$E_{flow} = \sum_{x \in \Lambda} \left| \nabla \psi_1(x)^T d(x;a) + \psi_2(x) - \psi_1(x) \right|^p ,$$

- This solution verifies when $p = 2$:

$$\frac{\partial E_{flow}}{\partial a} = 2 \sum_{x \in \Lambda} (\nabla \psi_1(x)^T d(x;a) + \psi_2(x) - \psi_1(x)) \frac{\partial d(x;a)}{da} \nabla \psi_1(x)$$

- However, the optical flow equation is valid only when the motion is small
- In practice, we use the DFD error criterion, and find the minimal solution using the gradient descent or exhaustive search method.

Motion estimation criteria IV

- About constant intensity assumption: minimizing the DFD error or solving the optical flow equation does not always give physically meaningful motion estimate.
- This is partially because the constant intensity assumption is not always correct.
- Regularization: the ME problem is ill-defined, and then to obtain physically meaningful solutions, one need to impose additional constraints to regularize the problem.
- To this aim, we can add a penalty term in our equation to enforce the smoothness of our vector field (i.e. it must vary smoothly across space except at boundaries):

$$E_s = \sum_{x \in \Lambda} \sum_{y \in N_x} \|d(x; a) - d(y; a)\|^2$$

where N_x is the set of adjacent pixels of x,

Motion estimation criteria V

- Then we want to minimize:

$$E_{total} = E_{DFD} + w_s\ E_s$$

  with w the weighting coefficient.

- We have to regularize but not too much! (to avoid over-blurring).

Minimization methods

Minimization methods I

- Minimization methods are various, but we will see only exhaustive search and gradient-based search methods.
- Usually, for the exhaustive search, the MAD is used for reasons of computational complexity, whereas for the gradient-based search, the ME is used for its mathematical tractability (we can compute the derivation !).
- The exhaustive search method guarantees reaching the global minimum.
- However, such search is feasible only if the number of unknown parameters is small, and each parameter takes only a finite set of discrete values.
- To reduce the search time, various fast algorithms can be developed, which achieve sub-optimal solutions.
- The most common gradient descent methods include the steepest gradient descent and the Newton-Raphson method.
- A gradient descent method can handle unknown parameters in a high dimensional continuous space.

Minimization methods II

- However, it can only guarantee the convergence to a local minimum.
- The error functions used in video processing are usually not convex and can have many local minima that are far from the global minimum.
- Therefore, it is important to obtain a good initial solution through the use of a prior knowledge, or by adding a penalty term to make the error function convex.

Minimization methods III

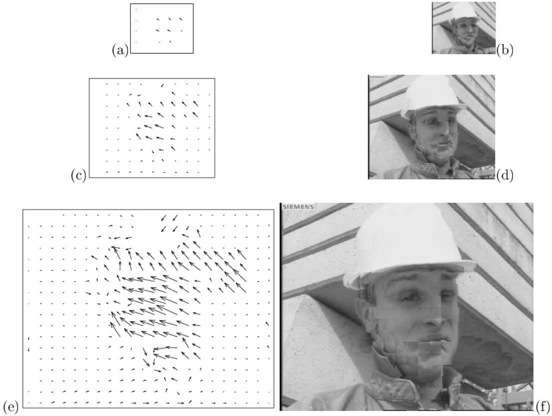


Figure 6.21. Example motion estimation results by HBMA for the two images shown in Fig. 6.8: (a-b) the motion field and predicted image at level 1; (c-d) the motion field and predicted image at level 2; (e-f) the motion field and predicted image at the final level (PSNR=29.32); A three-level HBMA algorithm is used. The block size is 16×16 at all levels. The search range is 4 at all levels with integer-pel accuracy. The result in the final level is further refined by a half-pel accuracy search in the range of ± 1.

Minimization methods IV

- One important search strategy is to use a multi-resolution representation of the motion field and conduct the search in a hierarchical manner.
- The basic idea is too first search the motion parameters in a coarse resolution, propagate this solution into a finer resolution, and then refine the solution in the finer resolution.
- It can combat both the slowness of exhaustive search methods and the non-optimality of gradient-based methods.

Regularization using motion smoothness constraint

Regularization using motion smoothness constraint I

- Since pixel-based motion representation is ill-defined, we need regularization techniques.
- Horn and Schunck propose to estimate the motion vectors by minimizing the following energy function:

$$E = \sum_{x \in \Lambda} \left(\frac{\partial \psi}{\partial x} v_x + \frac{\partial \psi}{\partial y} v_y + \frac{\partial \psi}{\partial t} \right)^2 + w_s \left(\|\nabla v_x\|^2 + \|\nabla v_y\|^2 \right)$$

- Interpretation: it is a combination of the flow-based criterion (we want it to be as much as possible close to 0) and a motion smoothness criterion (the more the motion field is smooth, the more it will tend to 0).
- Trick : in order to avoid over-smoothing of the motion field, Nagel suggest an oriented-smoothness constraint (smoothness along the object boundaries, but not across the boundaries).

Block matching algorithm (BMA)

Block matching algorithm (BMA) I

- One way to impose some smoothness constraints on the estimated motion field is to divide the image domain into non-overlapping small regions, called block
- We assume that the motion within each block can be characterized by a simple parametric model (constant, affine, or bilinear).
- If the block is sufficiently small, then this model can be quite accurate.
- We will use B_m to represent the m-th image block, M the number of blocks, and $\mathcal{M} = \{1, 2, \ldots, M\}$.
- The partition into blocks should satisfy:

$$\bigcup_{m \in \mathcal{M}} B_m = \Lambda,$$

and

$$B_m \bigcap B_n = \emptyset, \; m \neq n$$

Block matching algorithm (BMA) II

- Theoretically, a block can have any polygonal shape, but in practice, the square shape is used almost exclusively.
- In the simplest case, the motion in each block is assumed to be constant (block-wise translational model).
- This kind of algorithm is referred as Block Matching Algorithm (BMA).

The exhaustive search block matching algorithm (EBMA)

The exhaustive search block matching algorithm (EBMA) I

- Given an image block in the reference frame B_m, the motion estimation problem at hand is to determine a matching block B'_m in the current frame so that the error between these two blocks is minimized.
- The displacement vector d_m between the spatial positions of these two blocks (the center or a selected corner) is the MV of this block.
- Under the block-wise translational model,

$$w(x; a) = x + d_m, \; x \in B_m,$$

- Then the error can be written:

$$E(d_m, \forall m \in \mathcal{M}) = \sum_{m \in \mathcal{M}} \sum_{x \in B_m} |\psi_2(x + d_m) - \psi_1(x)|^p$$

The exhaustive search block matching algorithm (EBMA) II

- We can estimate the MV for each block individually:

$$E_m(d_m) = \sum_{x \in B_m} |\psi_2(x + d_m) - \psi_1(x)|^p$$

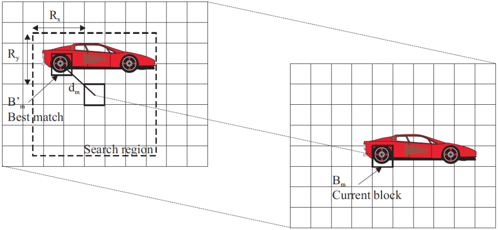


Figure 6.6. The search procedure of the exhaustive block matching algorithm.

- One way to determine the d_m that minimizes the above error is by using Exhaustive block matching algorithm (EBMA) (see the figure above).

The exhaustive search block matching algorithm (EBMA) III

- The EBMA determines the optimal d_m for a given block B_m in the reference frame (error minimization).
- To reduce the computational load, the MAD error ($p = 1$) is often used.
- In 2002, the computation of the EBMA needed VLSI/ASIC hardware architectures since no software were not able to do so in real-time.

[SEE NOTEBOOK]

Deformable block matching algorithm (6.5)

Introduction I

- In the BMA introduced previously, each block is assumed to undergo a pure translation,
- This model is not appropriate for blocks undergoing rotation, zooming, ...
- In general, a more sophisticated model, such as the affine, bilinear, or projective mappin, can be used to describe the motion of each block.
- Obviously, it will still cover the translational model as a special case.

Introduction II

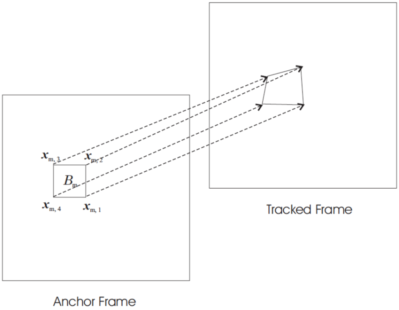


Figure 6.11. The deformable block matching algorithm finds the best matching quadrangle in the tracked frame for each block in the anchor frame. The allowed block deformation depends on the motion model used for the block. Adapted from [39, Fig. 6.9].

- With such models, a block in the anchor frame is in general mapped to a non-square quadrangle.
- Therefore, we refer to the case of bloack-based motion estimation methods using higher order models as deformable block-matching algorithm (DBMA).

Introduction III

- It is also known as generalized block-matching algorithm.
- In the following, we first discuss how to interpolate the MV at any point in a block using only the MVs at the block corners (called nodes), and then we present an algorithm for estimating nodal MVs.

Node-based motion representation I

- The most general model, the projective mapping, can be approximated by a polynomial mapping of different orders,
- Here, we introduce a node-based block motion model, which can characterize the same type of motions as the polynomial model, but is easier to interpret and to specify.
- In this model, we assume that a selected number of control nodes in a block can move freely and that the displacement of any interior point can be interpolated from nodal displacements.

Node-based motion representation II

- Let the number of control nodes be denoted by K, and the MVs of the control nodes in B_m by $d_{m,k}$, the motion function over the block is described by:

$$d_m(x) = \sum_{k=1}^{K} \phi_{m,k}(x) d_{m,k}, \; x \in B_m$$

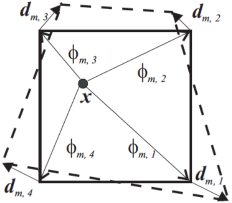


Figure 6.12. Interpolation of motion in a block from nodal MVs.

- The above equation expresses the displacement at any point in a block as an interpolation of nodal displacements.

Node-based motion representation III

- The interpolation kernel $\phi_{m,k}(x)$ depends on the desired contribution of the k-th control point in B_m to x.
- One way to designe the interpolation kernels is to use the shape functions (see next section) associated with the corresponding nodal structure.
- The translational, affine, and bilinear models are special cases of the node-base model with 1, 3, and 4 nodes, respectively.
- A model with more nodes can characterize more complex deformation.

Node-based motion representation IV

- The interpolation kernel in the 1-node case (at the block center or a chosen corner) is a pulse function, corresponding to nearest neighbor interpolation.
- The interpolation functions in the 3-node case (any three corners in a block) and 4-node (the four corners) cases are affine and bilinear functions respectively.
- Usually, to use an affine model with a rectangular block, the block is first divided into two triangles, and then each triangle is modeled by the 3-node model.

Node-based motion representation V

- The nodal MVs can be estimated more easily and specified with a lower precision than for the polynomial coefficients:
- First, it is easy to determine appropriate search ranges and search stepsizes for the nodal MVs than the polynomial coefficients, based on the a priori knowledge about the dynamic range of the underlying motion and the desired estimation accuracy.
- Second, all the motion parameters in the node-based representation are equally important, while those in the polynomial representation cannot be treated equally.

! For example, the estimation of the high order coefficients is much harder than the constant terms.

- Finally, specification of the polynomial coefficients requires a high order of precision: a small change in a high order can generate a very different motion field.

Motion estimation using node-based model I

- Notation: Because the estimation of nodal movements are independent from block to block, we skip the subscript m which indicates which block is being considered.
- The following derivation applies to any block B.
- With the node-based motion model, the motion parameters for any block are the nodal MVs, i.e.,

$$\boldsymbol{a} = [d_k; k \in \mathcal{K}],$$

with $\mathcal{K} = \{1, 2, \ldots, K\}$.

Motion estimation using node-based model II

- They can be estimated by minimizing the prediction error over this block, i.e.,

$$E(\boldsymbol{a}) = \sum_{\boldsymbol{x} \in B} (\psi_2(w(\boldsymbol{x}; \boldsymbol{a}) - \psi_1(\boldsymbol{x})))^2$$

  where

$$w(\boldsymbol{x}; \boldsymbol{a}) = \boldsymbol{x} + \sum_{k \in \mathcal{K}} \phi_k(\boldsymbol{x})\, d_k$$

- As with the BMA, there are many ways to minimize the error, including the exhaustive search and the gradient-based search method.
- The computational load required for the exhaustive search, however, can be unacceptable in practice, because of the high dimension of the search space.
- Gradient-based methods are more feasible in this case.

Motion estimation using node-based model III

- Usually we use the Newton-Raphson, but here we will use, for sake of simplicity, the finite difference method.
- For the derivatives approximations of the function $\boldsymbol{x} \rightarrow f(\boldsymbol{x})$:

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_i} \approx \frac{f(\boldsymbol{x} + h\ \boldsymbol{x}_i) - f(\boldsymbol{x} - h\ \boldsymbol{x}_i)}{2h}$$

- For the parameters updates using the gradient descent:

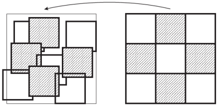
$$\boldsymbol{a}_i^{l+1} = \boldsymbol{a}_i^l - \alpha \frac{\partial E}{\partial \boldsymbol{a}_i}(\boldsymbol{a}^l)$$

- A good initial solution can often be provided by the EBMA.

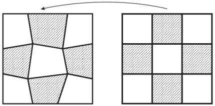
Mesh-based motion estimation (6.6)

Introduction I

- With the block-based model used either in BMA or DBMA, motion parameters in individual blocks are independently specified.



(a)



(b)

Introduction II

- Unless motion parameters of adjacent blocks are constrained to vary smoothly, the estimated motion field is often discontinuous and sometimes chaotic (see (a) in the figure above).
- One way to overcome this problem is by using mesh-based motion estimation.
- As illustrated in the figure above (see (b)), the anchor frame is covered by a mesh, and the motion estimation problem is **to find the motion of each node** (so that the image pattern within each element in the anchor frame matches well with that in the corresponding deformed element in the tracked frame).

Introduction III

- The motion within each element is interpolated from nodal's MVs.
- As long as the nodes in the tracked frame still form a feasible mesh (no self-crossing), the mesh-based motion representation is guaranteed to be continuous
  → thus it is from the blocking artifacts associated with block-based representation.
- Another benefit of the mesh-based representation is that it enables continuous tracking of the same set of nodes over consecutive frames, which is desirable in applications require object tracking.

Introduction IV

- Note that the inherent continuity with the mesh-based representation is not always desired.
- In real-world video sequences, there are often motion discontinuities at object boundaries.
- A more accurate representation would be to use separate meshes for different objects.

Introduction V

- As with the block-based representation, the accuracy of the mesh-based representation depends on the number of nodes.
- A very complex motion field can be reproduced as long as a sufficient number of nodes are used.
- To minimize the number of nodes required, the mesh should be adapted to the imaged scene, so that the actual motion within each element is smooth (i.e. can be interpolated accurately from the nodal motions).

Mesh-based motion representation I

- With the mesh-based motion representation, the underlying image domain in the anchor frame is partitioned into non-overlapping polygonal elements.

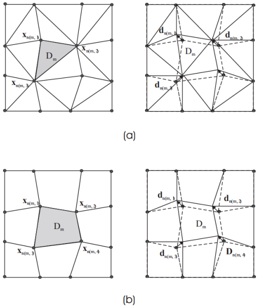


Figure 6.14. Illustration of mesh-based motion representation: (a) using a triangular mesh, with 3 nodes attached to each element, (b) using a quadrilateral mesh, with 4 nodes attached to each element. In the shown example, the two meshes have the same number of nodes, but the triangular mesh has twice the number of elements. The left column shows the initial mesh over the anchor frame, the right column the deformed mesh in the tracked frame.

Mesh-based motion representation II

- Such a mesh is also known as a control grid.
- In the mesh-based representation, the motion field over the entire frame is described by MVs at the nodes only.
- The MVs at the interior points of an element are interpolated from the MVs at the nodes of this element.
- The nodal MVs are constrained so that the nodes in the tracked frame still form a feasible-mesh with no flip-over elements.

Mesh-based motion representation III

- Notations:
  - Let the number of elements and nodes be denoted by M and N respectively, and the number of nodes defining each element by K.
  - $\mathcal{M} = \{1, 2, \dots, M\}$ (set of elements),
  - $\mathcal{N} = \{1, 2, \dots, N\}$ (set of nodes),
  - $\mathcal{K} = \{1, 2, \dots, K\}$ (set of numbers of nodes),
  - Let the m-th element and n-th node in frame t ($t = 1$ for the anchor frame and $t = 2$ for the tracked frame) be denoted by $B_{t,m}$, $m \in \mathcal{M}$, and $x_{t,n}$, $n \in \mathcal{N}$, and the MV of the n-th node by $d_n = x_{2,n} - x_{1,n}$.

Mesh-based motion representation IV

- The motion field in element $B_{1,m}$ is related to the nodal MVs d_n by:

$$d_m(x) = \sum_{k \in \mathcal{K}} \phi_{m,k}(x) d_{n(m,k)}, \; x \in B_{1,m},$$

  where $n(m,k)$ specifies the global index of the k-th node in the m-th element (cf. Figure 6.14).

- The function $\phi_{m,k}(x)$ is the interpolation kernel associated with node k in element m.
- It depends on the desired contribution of the k-th node in $B_{1,m}$ to the MV at x.

Mesh-based motion representation V

- To guarantee continuity across element boundary, the interpolation kernels should satisfy:

$$0 \leq \phi_{m,k}(x) \leq 1, \ \forall x \in B_m,$$

  and

$$\sum_k \phi_{m,k}(x) = 1, \ \forall x \in B_m,$$

  and

$$\phi_{m,k}(x_l) = \delta_{k,l} \quad (\delta \text{ is as usual the Kronecker})$$

- In other words, only a part of the nodes are used for each polygonal element.
- The interpolation kernels are called shape functions in finite element method (FEM).
- If all the polygonal elements have the same shape, then all the interpolation kernels are equal, that is, $\phi_{m,k}(x) = \phi_k(x)$.

Mesh-based motion representation VI

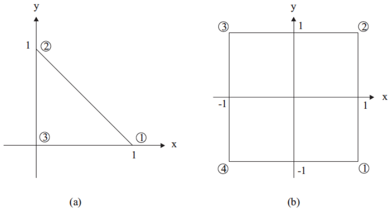


Figure 6.15. (a) A standard triangular element; (b) A standard quadrilateral element (a square).

- Standard triangular and quadrilateral elements are shown in Figure 6.15.

Mesh-based motion representation VII

- The interpolation kernels for the standard triangular element are:

$$\phi_1^t(x,y) = x,$$
$$\phi_2^t(x,y) = y,$$
$$\phi_3^t(x,y) = 1 - x - y.$$

Mesh-based motion representation VIII

- The interpolation kernels for the standard quadrilateral element are:

$$\phi_1^q(x,y) = (1+x)(1-y)/4,$$
$$\phi_2^q(x,y) = (1+x)(1+y)/4,$$
$$\phi_3^q(x,y) = (1-x)(1+y)/4,$$
$$\phi_4^q(x,y) = (1-x)(1-y)/4.$$

Mesh-based motion representation IX

- We see that the interpolation kernels for these two cases are affine and bilinear functions respectively.

! The coefficients of these functions depend on the node position.

- Note that here the nodes and elements are determined using global indices.
- This is necessary because the nodal MVs are not independent from polygonal element to polygonal element.

! It is important not to confuse nodal- and mesh-based models: in the nodal one, although several adjacent blocks can share the same node, the nodal MVs are determined independently in each block.

Mesh-based motion representation X

- In the mesh-based model, node n is assigned a single MV, which will affect the interpolated motion functions in four quadrilateral elements attached to this node.
- In node-based model, node n can have four MVs, depending on in which block it is considered.

Motion estimation using mesh-based method I

- With the mesh-based motion representation, the motion parameters include the nodal MVs, that is,

$$\boldsymbol{a} = \{d_n, n \in \mathcal{N}\},$$

- To estimate them, we can again use an error minimization approach (see the nodal case).

- Under the mesh-based motion model, the DFD error becomes:

$$E(d_n, n \in \mathcal{N}) = \sum_{m \in \mathcal{M}} \sum_{x \in B_{1,m}} |\psi_2(w_m(x)) - \psi_1(x)|^p,$$

where

$$w_m(x) = x + \sum_{k \in \mathcal{K}} \phi_{m,k}(x) d_{n(m,k)}, \; x \in B_{1,m}.$$

Global motion estimation

Introduction I

- Depending on the context, the mapping function between the two images can be described by a translation, a geometric transformation, an affine mapping, or a projection mapping.
- There are usually at least two objects, a stationnary background and one or more moving foregrounds.
- Fortunately, when the foreground object motion is small and the camera does not move in the Z-direction, the motion field can be approximated well by a global model.

Introduction II

- Such camera motions are quite common in sports videos and movies.
- As long as the effect of the camera motion dominates over other motions (motion of individual small objects), determination of this dominant global motion is still very useful.
- Two approaches are possible for estimating the global motion:
  - minimization of the prediction error,
  - first determine (pixel-/block-wise) MVs, and then using a regression method.

Region-based motion estimation

Region-based motion estimation I

- There are usually multiple types of motions in the imaged scene,
- By region-based motion estimation, we mean to segment the underlying image frame into multiple regions and estimate the motion parameters of each region.
- The segmentation should be such that a single parametric motion model can represent well the motion in each region.
- The simplest approach is to require each region undergo the same translational motion.
- For a more efficient motion representation, an affine or bilinear or perspective motion model should be used.

Region-based motion estimation II

- 3 possible approaches:
  (1) One first segments the image frame into different regions based on texture homogeneity, the edge information, and then estimates the motion in each region.

  !! We call such a method region-first.

  (2) One first estimates the motion field over the entire frame, and then segments the resulting motion field so that the motion in each region can be modeled by a single parameter model.

  ! We call this model motion first.

  !! The resulting region can be further refined subject to some spatial connectivity constraints.

Region-based motion estimation III

!!! The second problem involves motion-based segmentation.

(3) One jointly estimates region partition and motion in each region.

! In general, this is accomplished by an iterative process, which performs region segmentation and ME alternatively.

Multi-resolution motion estimation

Multi-resolution motion estimation I

- Various ME approaches can be reduced (see below) to solving an error minimization problem.
- Major difficulties:
  - many local minima in the gradient-descent case,
  - it is not easy to reach the global minimum except if we are close to the initial solution,
  - the amount of computation involved in the minimization process is very high.
- A good alternative is then the multi-resolution approach!
- It searches the solution for an optimization problem in successively finer resolutions.
- By first searching the solution in coarse resolution, one can usually obtain a solution that is close to the true motion.
- The total number of searches is reduced (small neighborhoods in each scale).

Multi-resolution motion estimation II

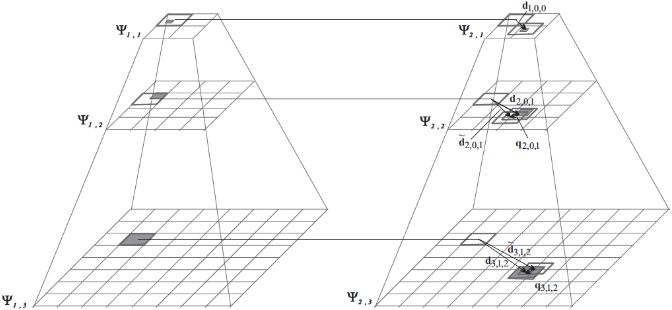


Figure 6.19. Illustration of the Hierarchical Block Matching Algorithm.

Figure: A multi-resolution approach (HBMA)

Multi-resolution motion estimation III

- The benefits of the multi-resolution are two folds:
  - First, the minimization problem at a coarser resolution is less ill-posed than at a finer resolution,
  - The solution obtained at a coarser level is closer to the true solution at that level,
  - At the end, we converge then more easily to the global minimum.
  - Second, the estimation at each resolution level can be confined to a smaller range,
  - Then the total number of searches is smaller than in the other methods.
- The use of multi-resolution representation for image processing was first introduced by Burt and Adelson (Laplacian Pyramid).

References

Indrajit Chakrabarti, Kota Naga Srinivasarao Batta, S. K. C. (2015).
Motion Estimation for Video Coding.
Springer.

Jie Chen, Ut-Va Koc, K. R. L. (2002).
Design of Digital Video Coding Systems.
Signal Processing and Communication Series, Marcel Dekker, Inc.

Yao Wang, Jorn Ostermann, Y.-Q. Z. (2002).
Video Processing and Communications.
Prentice Hall.