Image Segmentation, Part 2

Lecture 9

See Section 5.2 in
Reinhard Klette: Concise Computer Vision
Springer-Verlag, London, 2014

---

\(^1\) See last slide for copyright information.
Agenda

1. Mean-Shift Segmentation
2. Mean-Shift Model
3. Non-Optimised Algorithm
4. Time Optimization
Feature Space for Mean-Shift Segmentation

*Mean-shift segmentation* operates in image feature space

For a color image we may use, e.g., the 3D RGB feature space

A gray-level image defines only a 1D feature space of values $I(p)$

For generating an $n$-channel image for segmentation, with $n \geq 1$, we can also use the 2 channels $I_x(p)$ and $I_y(p)$ of

$$\nabla I(p) = \text{grad } I(p) = \left[ \frac{\partial I}{\partial x}(p), \frac{\partial I}{\partial y}(p) \right]^\top$$

possibly also combined with 2 more channels for mean and standard deviation in local neighborhoods

i.e. (in general) channels may represent local properties at pixel location $p$
A Way for Calculating Means in Feature Spaces

**Given:** Scalar image \( I \) and region \( S \subset \Omega \)

**Moments of** \( S \) **in** \( I \) **of order** \( a + b \), for \( a, b \geq 0 \):

\[
m_{a,b}(S) = \sum_{(x,y) \in S} x^a y^b \cdot I(x, y)
\]

If \( I(x, y) = 1 \) in \( S \) then area \( \mathcal{A}(S) = m_{0,0}(S) = \sum_{(x,y) \in S} I(x, y) = |S| \)

**Centroid** \((x_S, y_S)\) defined by

\[
x_S = \frac{m_{1,0}(S)}{m_{0,0}(S)} \quad \text{and} \quad y_S = \frac{m_{0,1}(S)}{m_{0,0}(S)}
\]

based on moments of order 1:

\[
m_{1,0}(S) = \sum_{(x,y) \in S} x \cdot I(x, y) \quad \text{and} \quad m_{0,1}(S) = \sum_{(x,y) \in S} y \cdot I(x, y)
\]
Now: Means for $nD$ Feature Spaces

In generalisation: Assume an $n$-dimensional feature histogram $H$
(i.e. multiplicities of feature vectors in image $I$)

Components of $nD$ mean of a set $S$ of features $u = (u_1, \ldots, u_n)$ defined by

$$u_{i,S} = \frac{m_{0,\ldots,0,1,0,\ldots,0}(S)}{m_{00\ldots0}(S)}, \quad \text{for} \quad i \in \{1, \ldots, n\}$$

using moments $m_{0,\ldots,0,1,0,\ldots,0}(S)$ (with the 1 in the $i$th position) and
$m_{00\ldots0}(S)$ (with $n$ zeros) of multiplicities of feature vectors (i.e. use of
histogram $H$ in defining sums); e.g.:

$$m_{0,\ldots,0,1,0,\ldots,0}(S) = \sum_{(u_1,\ldots,u_n)\in S} u_i \cdot H(u_1, \ldots, u_n)$$
Example: \( n = 2 \)

Gray-level image Odense with \( G_{\text{max}} = 255 \), and gray-level visualization of a 2D feature space for this image defined by mean and standard deviation in \( 3 \times 3 \) pixel neighborhoods; local maxima are expected for small values of \( \sigma \).

Use of a color key for visualising multiplicities of feature vectors.
Mean-Shift in Feature Space

A $5 \times 5$ image $I$ with $u = (a, b)$, for $0 \leq a, b \leq 7$

Thus: 25 feature pairs in the $ab$ feature space (a 2D histogram)

Center of circular window moves into the mean position

Figure illustrates two mean-shift moves

This is not the image plane; image not shown here

Numbers 1, 2 or 3 are multiplicities of occurring feature pairs
Iteration and Stop Criterion

Select pair \((2, 4)\) as initial mean which has multiplicity 2.

Calculate mean of set \(S\) defined by radius \(r = 1.6\) around this pair.

Move circular window (defining current set \(S\)) such that center now at new mean.

New mean \(\mathbf{u}_S = (u_{1, S}, u_{2, S})\) for \(S\) again.

Iteration stops if distance between subsequent means below threshold \(\tau\) (e.g. \(\tau = 0.1\)).

We have the final mean \(\mathbf{u}_S\) (reference point), assumed to be close to a local maximum in feature space.

Pixels having feature value \((2, 4)\) are assigned to final mean \(\mathbf{u}_S\) (defining a set of segments in the image).
Mean-Shift Segmentation

Repeat mean-shift procedure by starting at any feature vector

Each time a final mean

Final means being very close to each other are clustered (identified)

Pixels having feature values which are assigned to final means in the same cluster define connected regions being the *segments*

Each segment can be labeled by one unique label (integer, color, name, ...)

(This is just a start into the area of various, more specific mean-shift segmentation algorithms)
Mean-Shift is a Steepest-Ascent Method

Consider feature space multiplicities as elevations
Shifting window moves up the hill
Reference point finally (roughly) at a local peak (i.e. a local maximum)

Method depends on parameters $r$ and $\tau$
Local Peak and Global Mode

*Mode* is the value that occurs most often in a set of data

**Example:** mode of set \{12, 15, 12, 11, 13, 15, 11, 15\} is 15

Set of data may have several modes

**Example:** set \{12, 15, 12, 11, 13, 10, 11, 14\} is *bimodal*

Set \{16, 15, 12, 17, 13, 10, 11, 14\} has no mode

A local peak is not necessarily a global mode; global mode(s) define(s) local peak(s)
Agenda

1. Mean-Shift Segmentation
2. Mean-Shift Model
3. Non-Optimised Algorithm
4. Time Optimization
Kernel Function and Profile

For generalizing the circular window, consider a rotation-symmetric
kernel function

\[ K(u) = c_k \cdot k(\|u\|_2^2) \]

defined at feature points \( u \) by a 1D profile \( k \) and constant \( c_k \)

\[ \int_{\mathbb{R}^n} K(u) \, du = 1 \]

Kernel defines weights, similar to local convolutions in images
but now in feature space

Here: only a brief sketch about the underlying math
(see book for more details)
Examples of Profiles

Profiles $k(a)$, for $a \in \mathbb{R}$

Epanechnikov function:

$$k(a) = \frac{3}{4}(1 - a^2) \quad \text{for} \quad -1 < a < 1 \text{ and } 0 \text{ elsewhere}$$
Density Estimator

Radius $r > 0$ of the kernel (e.g. $\sigma$ for Gauss function)

Assume $m$ feature vectors $\mathbf{u}_i$, $1 \leq i \leq m$, in $\mathbb{R}^n$

For an image $I$ we have $m = N_{\text{cols}} \cdot N_{\text{rows}}$

Density estimator at feature vector $\mathbf{u}$ for kernel $K$:

$$f_k(\mathbf{u}) = \frac{1}{mr^n} \sum_{i=1}^{m} K \left( \frac{1}{r} \cdot (\mathbf{u} - \mathbf{u}_i) \right)$$

Corollary (see proof in book):

Mean-shift proceeds in direction of $\text{grad} \ f_k(\mathbf{u})$

(i.e. partial derivatives of $f_k$ w.r.t. $u_i$, $1 \leq i \leq n$)

towards a local peak defined by $\text{grad} \ f_k(\mathbf{u}) = 0$
Agenda

1. Mean-Shift Segmentation
2. Mean-Shift Model
3. Non-Optimised Algorithm
4. Time Optimization
Algorithm Using Linear Algebra

Algorithm without any time optimization

Data as on Frame 7 represent

\[
\{ (5, 2), (6, 2), (3, 3), (3, 3), (4, 3), \\
(5, 3), (1, 4), (2, 4), (2, 4), (4, 4), \\
(5, 4), (5, 4), (5, 4), (6, 4), (7, 4), \\
(2, 5), (3, 5), (3, 5), (5, 6), (6, 6), \\
(6, 6), (3, 7), (4, 7), (6, 7), (6, 7) \}
\]

assuming that \( a \) and \( b \) are integers starting at 1

Data matrix

\[
\begin{bmatrix}
5 & 6 & 3 & 3 & 4 & 5 & 1 & 2 & 2 & 4 & 5 & 5 & 5 & 6 & 6 & 7 & 2 & 3 & 3 & 5 & 6 & 6 \\
2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 5 & 5 & 5 & 6 & 6 & 7 \\
\end{bmatrix}
\]
Participating Matrices

Select a feature pair: one of the two pairs (3, 5)

Mean matrix $\mathbf{M}$ with (3, 5) in all of its columns

$$
\begin{bmatrix}
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5
\end{bmatrix}
$$

Calculate squared Euclidean distances between columns in $\mathbf{M}$ and columns in $\mathbf{D}$ by taking squares of differences $\mathbf{D} - \mathbf{M}$

(Squared) Euclidean distance matrix

$$
\begin{bmatrix}
4 & 9 & 0 & 0 & 1 & 4 & 4 & 1 & 1 & 1 & 4 & 4 & 4 & 9 & 16 & 1 & 0 & 0 & 4 & 9 & 9 & 0 & 1 & 9 & 9 \\
9 & 9 & 4 & 4 & 4 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 4 & 4 & 4 & 4
\end{bmatrix}
$$

Corresponding sums in each column

$$
\begin{bmatrix}
18 & 4 & 4 & 5 & 8 & 5 & 2 & 2 & 2 & 5 & 5 & 5 & 10 & 17 & 1 & 0 & 0 & 5 & 10 & 10 & 4 & 5 & 13 & 13
\end{bmatrix}
$$
Iteration and Stop

Sums need to be compared with $r^2$

Let $r = 1.6$, thus $r^2 = 2.56$, assume uniform profile

Six squared Euclidean distance values are less than 2.56

These are the six feature vectors in the circle defining the currently active set $S = \{(2, 4), (2, 4), (4, 4), (2, 5), (3, 5), (3, 5)\}$

Calculate mean of $S$ as $\mathbf{u}_S = (2.67, 4.5)$

Compare $(2.67, 4.5)$ with previous mean $(3, 5)$; distance still above $\tau = 0.1$

Next iteration step:
new matrix $\mathbf{M}$ which has $(2.67, 4.5)$ in all of its columns ETC.

Stop if distance between two subsequent means less than $\tau$
Mean-Shift Math.Model Algorithm Time Optimization

Agenda

1. Mean-Shift Segmentation
2. Mean-Shift Model
3. Non-Optimised Algorithm
4. Time Optimization
Partitioning the Feature Space into Windows

Partition uniformly-digitized feature space into windows

Perform mean-shift (just) for all windows, until no window moves anymore into a new position

Round means to nearest integers

Assign feature vectors to final means

**Example:** Assume uniform profile in windows

Blue: stop of a move
Red: a second window moves into the same final mean
Partitioned Feature Space

Partitioned feature space for image Odense and segmentation result
### Possible Cases (Here: for Smaller Windows)

#### Mean-shift moves of $2 \times 2$ windows

- **Gray:** Irrelevant window
- **Blue:** No further move
- **Red:** Merger with a blue cell
- **Black:** Merger with a red cell
- **Yellow:** A previously already considered location - the most interesting case for time optimization
Resulting Clusters in Feature Space

Initial $2 \times 2$ partitioning

Final locations of means with indicated 4- or 8-adjacencies

Resulting equivalence classes, also merged
  if clustering by 4-connectedness
  if clustering by 8-connectedness
Example 1

Color image HowMany. Segmentation result using a 3D RGB feature space, a partitioning into $3 \times 3$ windows, without any clustering of final means.
Example 2

Segmentation result for image How Many using a 3D RGB feature space, a mean-shift algorithm for individual feature vectors, and the Gauss kernel function with $r = 25$. All the same as before, but with merging final means which are in a distance of $\leq 50$ to each other.
Copyright Information

This slide show was prepared by Reinhard Klette with kind permission from Springer Science+Business Media B.V.

The slide show can be used freely for presentations. However, all the material is copyrighted.


In case of citation: just cite the book, that’s fine.