Depth map calculation for a variable number of moving objects using Markov sequential object processes

M.N.M. van Lieshout

collette@cwi.nl

CWI & Eindhoven University of Technology
The Netherlands

Research sponsored by the Technology Foundation STW, applied science division of NWO, and the technology programme of the Ministry of Economic Affairs.

Thanks to: A.G. Steenbeek and C. Varekamp.
Motivation: 3D television

Catch 22:

- dearth of ‘true’ 3D content prevents consumers from buying a 3D screen;
- dearth of consumers with a 3D screen discourages producers to film in 3D.

Temporary solution: transform plentiful 2D content to a format suitable for display on a 3D television by adding depth.
Depth clues from motion

When objects pass each other, their image projections overlap so their relative distance to the camera can be determined and propagated over frames.

A tracking algorithm must:

- decide whether there are any objects of a specified kind in each of the image frames;
- if so, determine the number of objects, their locations, shapes, sizes;
- any spatial relationship between objects within a frame;
- object movements across frames (in which frame is it first seen, its trajectory over time, and the frame where the object is last observed).
Model ingredient I: data

Sequence of $I \in \mathbb{N}$ images

$$y = (y^i; i = 1, \ldots, I)$$

where each image $y^i$ is a collection of pixel values

$$y^i = (y^i_t; t \in T), \quad T \text{ finite.}$$

The observed values $y^i_t$ range over some set $V$, e.g. $\{0, 1, \ldots, 255\}^d$ with $d = 1$ for grey level and $d = 3$ for colour.
**Model ingredient II: objects**

The **object space** is a Cartesian product

\[ D \times L \]

with \( D \subset \mathbb{R}^2 \), compact, to specify location, and \( L \) an arbitrary Polish space for shape, size, colour, ... Each object \( x \in D \times L \) leaves a footprint \( R(x) \subseteq T \) in image space, its **template**.

**Example:** the template of a square parametrised by

\[ (x, y, s, v) \in \mathbb{R}^2 \times (\mathbb{R}^+ \times \{0, \ldots, 255\}) \]

is a square with top left pixel \( \lfloor x \rfloor, \lfloor y \rfloor \), bottom right \( \lceil x + s \rceil, \lceil y + s \rceil \), coloured \( v \), clipped to \( T \).
Model ingredient III: sequence of object scenes

An **object configuration** is a finite vector of objects

\[ \vec{x} = (x_1, \ldots, x_n) \]

where \( x_j \in D \times L, \ j = 1, \ldots, n, \ n \geq 0 \), which is mapped to a **signal** image

\[ \theta_t(\vec{x}) := \left\{ \begin{array}{ll} \theta_t(x_j) & \text{if } t \in R(x_j) \setminus \bigcup_{k<j} R(x_k) \\ \theta_0 & \text{if } t \in T \setminus \bigcup R(x_j) \end{array} \right. \]

using the templates and parameters \( \theta_t(x) \in \Theta \) compatible with \( V \) that represent the ideal (noise-free) image. Write \( \theta_0 \) for the background signal.

A sequence of object configurations is denoted by

\[ x = (\vec{x}^1, \ldots, \vec{x}^I). \]
Tracking as statistical inference problem

In order to quantify how well a sequence of object configurations describes a given video sequence, we formulate a Gibbs distribution

\[ f(x) \propto \exp [-U(x)] \]

whose energy is the sum of two terms:

- **regression** for fit to observation;
- **regularisation** for spatial and temporal coherence.

Then find the mode of \( f(x) \) to get the best description by a Monte Carlo approach.
Regression

Let $\Theta$ and $V$ be compatible ($L_p$ distance exists) and write

$$U(x) = \sum_{i=1}^{I} \lambda_i L_p(y^i, \theta(x^i))^p$$

for $p = 1$ (Laplacian noise) or 2 (white noise) and $\lambda_i > 0$.

Minimisation is equivalent to least absolute deviation ($p = 1$) or least squares regression ($p = 2$).

**Disadvantages:**

- non-uniqueness;
- no correlation between objects in adjacent frames;
- oversegmentation;

due to occlusion: adding extra objects ‘behind’ the signal of those closer to the camera does not affect $U$. 
Local dependence

The potential energy required for adding $\xi$ to $\vec{x}^i$ in last position

$$\lambda_i \sum_{t \in R(\xi) \setminus \cup_k R(x^i_k)} \left[ |y^i_t - \theta_t(\xi)|^p - |y^i_t - \theta^i_0|^p \right]$$

depends only on $R(\xi)$ and those $R(x^i_k)$ that overlap $R(\xi)$. If $\xi$ were added at position $k$,

$$\sum_{t \in R(\xi) \setminus \cup_{l < k} R(x^i_l)} \left[ |y^i_t - \theta_t(\xi)|^p - |y^i_t - \theta_t(\vec{x}^i)|^p \right]$$

does not depend on $R(x^i_l)$ with $l < k$ that do not overlap $R(\xi)$; the second term involves only those $R(x^i_l)$ with $l \geq k$ that overlap $R(\xi)$.

Conclusion: the regression part is a **Markov sequential object process** with respect to the overlapping objects relation.
**Regularisation I: Within frame interaction**

**Idea:** replace energy $U = V_1$ by $U := V_1 + V_2 + V_3$.

For $\beta \in \mathbb{R}, \gamma > 0$, the **Strauss** model is

$$V_2(\vec{x}) = \beta n(\vec{x}) + \gamma n_o(\vec{x})$$

where $n(\vec{x})$ is the length of $\vec{x}$, and $n_o(\vec{x})$ is the number of pairs $\{u, v\}$ in $\vec{x}$ for which $R(u) \cap R(v) \neq \emptyset$.

**Local dependence:** if we add $u$ to $\vec{x}$,

$$V_2((\vec{x}, u)) - V_2(\vec{x}) = \beta + \gamma \# \{x_i \in \vec{x} : R(u) \cap R(x_i) \neq \emptyset\}$$

depends only on $R(x_i)$ that overlap $R(u)$, in other words is Markov with respect to the overlapping objects relation.
Regularisation II: Propagation over frames

Write

\[ S_{m,n} = \{(M, N, \pi) : M \subseteq \{1, \ldots, m\}; N \subseteq \{1, \ldots, n\}; |M| = |N|\} \]

with \( m, n \in \mathbb{N}_0 \), and \( \pi : M \rightarrow N \) a bijection.
\[ V_3(\mathbf{x}^i, \mathbf{x}^{i+1}, s^{i,i+1}) = \sum_{l \in M(s^{i,i+1})} \tau(\mathbf{x}_l^i, \mathbf{x}_{\pi(s^{i,i+1})}^{i+1}(l)) + \sum_{l \notin M(s^{i,i+1})} \lambda(\mathbf{x}_l^i) \]
\[ + \sum_{l \notin N(s^{i,i+1})} \lambda(\mathbf{x}_l^{i+1}) + \sum_{x_l \sim x_k \in \mathbf{x}^i; l < k \in M(s^{i,i+1})} \rho \mathbf{1} \{ \pi(s^{i,i+1})(l) > \pi(s^{i,i+1})(k) \} \]
\[ + \sum_{x_l \sim x_k \in \mathbf{x}^{i+1}; l < k \in N(s^{i,i+1})} \rho \mathbf{1} \{ \pi^{-1}(s^{i,i+1})(l) > \pi^{-1}(s^{i,i+1})(k) \} \].

(discrete Markov transition probability kernel in frame-time) where

- \( \lambda(\cdot) > 0 \) penalises unmatched objects;
- \( \tau(\cdot, \cdot) > 0 \) is a symmetric dissimilarity measure;
- \( \rho \geq 0 \) forces a similar ranking in index between objects that overlap in one frame in other frames.
Metropolis–Hastings algorithm

Markov sequential spatial processes arise naturally as the limit distribution of a Markov chain that

- proposes many ‘small’ changes to a sequence of object configurations;
- accepts an update with a probability that depends on the improvement in energy.

Some care has to be taken to ensure that

- from any sequence of object configurations, one can reach any other such sequence with positive probability;
- there are no cycles;
- changes in dimension are properly handled;
- each change is easy to implement;
- enough types of changes are included for efficient exploration.
Example: birth of singly matched object

1. select a frame uniformly from \( \{1, \ldots, I\} \), say \( i \);
2. with probability \( 1/2 \), link to each of the adjacent frames, say \( i - 1 \);
3. select an object in frame \( i - 1 \) not yet matched to an object in frame \( i \) uniformly at random, say \( x^{i-1}_k \);
4. sample from probability kernel \( k(\xi|x^{i-1}_k) \), and insert \( \xi \) into \( \bar{x}^i \) at a uniformly chosen position \( j \) to obtain \( c_j(\bar{x}^i, \xi) \);
5. add the match between \( x^{i-1}_k \) and \( \xi \) to \( s^{i-1,i} \) and adjust the indices to obtain \( c_j(s^{i-1,i}, \xi, x^{i-1}_k) \) and \( c_j(s^{i,i+1}, \xi) \);
6. accept the move according to the Hastings ratio

\[
\frac{n(\bar{x}^{i-1}) - |M(s^{i-1,i})|}{|N(s^{i-1,i}) \setminus M(s^{i,i+1})| + 1} \frac{1}{k(\xi|x^{i-1}_k)} \times \exp \left[ -U(x'; s'; y) + U(x; s; y) \right]
\]

where \( x' \) differs from \( x \) only in frame \( i \) with \( \bar{x}^{i'} = c_j(\bar{x}^i, \xi) \), and \( s' \) is identical to \( s \) except for the matchings involving frame \( i \) which are replaced by \( c_j(s^{i-1,i}, \xi, x^{i-1}_k) \) and \( c_j(s^{i,i+1}, \xi) \).
Updates

The following updates may be used:

• birth of singly matched object;
• birth of doubly matched object;
• birth of unmatched object;
• death of singly matched object;
• death of doubly matched object;
• death of unmatched object;
• modification of permutation order;
• birth of match;
• death of match;
• modification of object.
Consider

- Laplacian noise;
- dissimilarity $\tau(x_1, x_2)$ based on squared Euclidean distances between top left points, the absolute differences in side length, and absolute grey level difference.

**Note:** the optimal templates are identical to the data; in the middle frame, a square is hidden behind the light one for temporal consistency. Matches are correctly found.
Results

Depth is quantified by pairwise probabilities $p_{ij}^k$ of object $i$ having a lower sequence index than object $j$.

The result for frame 1 is:

\[
\begin{bmatrix}
- & 0.50 & 0.63 & 1.00 & 0.45 & 0.45 & 0.80 & 0.80 \\
0.50 & - & 0.62 & 1.00 & 0.45 & 0.45 & 0.80 & 0.80 \\
0.37 & 0.38 & - & 0.75 & 0.33 & 0.34 & 0.67 & 0.66 \\
0.00 & 0.00 & 0.25 & - & 0.10 & 0.10 & 0.40 & 0.40 \\
0.55 & 0.55 & 0.67 & 0.90 & - & 0.50 & 0.83 & 1.00 \\
0.55 & 0.55 & 0.66 & 0.90 & 0.50 & - & 1.00 & 0.83 \\
0.20 & 0.20 & 0.33 & 0.60 & 0.17 & 0.00 & - & 0.50 \\
0.20 & 0.20 & 0.34 & 0.60 & 0.00 & 0.17 & 0.50 & - \\
\end{bmatrix}
\]
Application: table tennis

- ellipses with three shape parameters;
- discrete RGB space, equipped with equal weight mixture of data frame histograms;
- background colour from data histograms;
- Gaussian noise.
- dissimilarity $\tau(x_1, x_2)$ based on squared Euclidean distances between centres, the absolute differences in axes lengths and orientation (modulo $\pi$), and absolute differences in RGB space.
Results

Optimal configuration:

Data masked by annealed object sequence (left) and annealed object sequence overlaid upon the data (right).

**Note:** the relative depth is correctly passed on from the middle frames to the neighbouring ones. Without temporal cohesion energy terms, the both orderings of bat and ball in frames 1, 3 would have had probability 1/2.
Summary

We presented an application of Markov sequential object processes to the calculation of depth maps with a view to 3D-TV.

The model

- can handle a variable number of interacting objects;
- is able to cope with the occlusion caused by objects at different depths;
- maintains the identity of objects as well as their relative depth over consecutive video frames;
- ensures fit to the data.

The computational complexity of the model can be handled by a suitably designed Metropolis–Hastings algorithm.