Causal estimation of 3D structure and motion

Stefano Soatto
UCLA

VISION as a SENSOR

machine to INTERACT with the environment

NEED to estimate relative 3D MOTION

3D SHAPE (TASK)

REAL-TIME

CAUSAL processing

representation of SHAPE
(only supportive of representation of motion)

POINT-FEATURES

TRADEOFFS

	SFM	CORRESPONDENCE
LARGE BASELINE	EASY	HARD/IMPOSSIBLE

TRADEOFFS

	SFM	CORRESPONDENCE
LARGE BASELINE	EASY	HARD/IMPOSSIBLE
SMALL BASELINE	HARD/IMPOSSIBLE	EASY

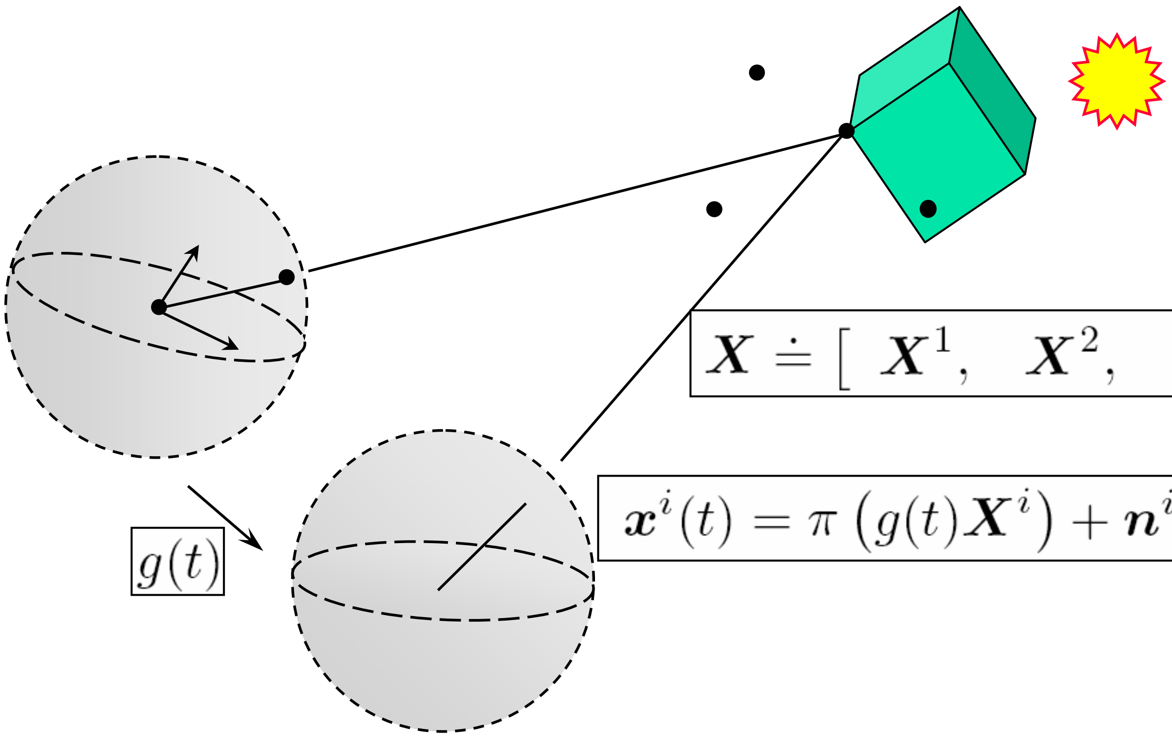
TRADEOFFS

	SFM	CORRESPONDENCE	
LARGE BASELINE	EASY	HARD/IMPOSSIBLE	GLOBAL
SMALL BASELINE	HARD/IMPOSSIBLE	EASY	LOCAL

WHAT DOES IT TAKE ?

	SFM	CORRESPONDENCE	
LARGE BASELINE	EASY	HARD/IMPOSSIBLE	GLOBAL
SMALL BASELINE	HARD/IMPOSSIBLE	EASY	LOCAL
	INTEGRATE visual information OVER TIME	OCCLUSIONS!	

Setup and notation



$$\boldsymbol{X} \doteq \left[\ \boldsymbol{X}^1, \quad \boldsymbol{X}^2, \quad \ldots, \quad \boldsymbol{X}^N \ \right] \quad \in \mathbb{R}^{3\times N}$$

$$\boldsymbol{x}^i(t) = \pi\left(g(t)\boldsymbol{X}^i\right) + \boldsymbol{n}^i(t) \quad \in \mathbb{R}^2$$

Temporal evolution:

$$\boldsymbol{X}(t+1) = \boldsymbol{X}(t)$$

$$g(t+1) = \exp\left(\widehat{\xi}(t)\right)g(t)$$

Structure and motion as a filtering problem

$$\begin{cases} \boldsymbol{X}(t+1) = \boldsymbol{X}(t), & \boldsymbol{X}(0) = \boldsymbol{X}_0 \in \mathbb{R}^{3\times N}, \\ g(t+1) = \exp\left(\widehat{\xi}(t)\right) g(t), & g(0) = g_0 \in SE(3), \\ \xi(t+1) = \xi(t) + \alpha(t), & \xi(0) = \xi_0 \in \mathbb{R}^6, \\ \boldsymbol{x}^i(t) = \pi\left(g(t)\boldsymbol{X}^i(t)\right) + \boldsymbol{n}^i(t), & \boldsymbol{n}^i(t) \sim \mathcal{N}(0, \Sigma_n). \end{cases}$$

- Given measurements of the "output" (feature point positions)
- Given modeling assumptions about the "input" (acceleration = noise)
- Estimate the "state" (3D structure, pose, velocity)

Difficulties …

$$\begin{cases} \boldsymbol{X}(t+1) = \boldsymbol{X}(t), & \boldsymbol{X}(0) = \boldsymbol{X}_0 \in \mathbb{R}^{3\times N}, \\ g(t+1) = \exp\left(\widehat{\xi}(t)\right) g(t), & g(0) = g_0 \in SE(3), \\ \xi(t+1) = \xi(t) + \alpha(t), & \xi(0) = \xi_0 \in \mathbb{R}^6, \\ \boldsymbol{x}^i(t) = \pi\left(g(t)\boldsymbol{X}^i(t)\right) + \boldsymbol{n}^i(t), & \boldsymbol{n}^i(t) \sim \mathcal{N}(0, \Sigma_n). \end{cases}$$

- Model is non-linear (output map = projection)
- State-space is non-linear! (SE(3))
- Noise: need to specify what we mean by “estimate”
- Even without noise: *model is not observable!*

Observability

- Equivalent class of state-space trajectories generate the same measurements

$$\boxed{\{\boldsymbol{X}_0, g_0, \xi_0\}} \qquad g_0 = (R_0, T_0) \;\; \xi_0 = (\omega_0, v_0)$$

$$\boxed{\{\beta \tilde{R} \boldsymbol{X}_0 + \beta \tilde{T}, \tilde{g}_0, \tilde{\xi}_0\}} \qquad \tilde{g}_0 = (R_0 \tilde{R}^T, \beta T_0 - \beta R_0 \tilde{R}^T \tilde{T}) \;\; \tilde{\xi}_0 = (\omega_0, \beta v_0)$$

- Fix, e.g., the direction of 3 points, and the depth of one point (Gauge transformation)

Local coordinatization of the state space

$$
\begin{cases}
\boldsymbol{x}_0^i(t+1) = \boldsymbol{x}_0^i(t), \quad i = 1, 2, \ldots, N, & \boldsymbol{x}_0^i(0) = \boldsymbol{x}_0^i, \\
\lambda^i(t+1) = \lambda^i(t), \quad i = 1, 2, \ldots, N, & \lambda^i(0) = \lambda_0^i, \\
T(t+1) = \exp(\widehat{\omega}(t))T(t) + v(t), & T(0) = T_0, \\
\Omega(t+1) = \log_{SO(3)}\big(\exp(\widehat{\omega}(t))\exp(\widehat{\Omega}(t))\big), & \Omega(0) = \Omega_0, \\
v(t+1) = v(t) + \alpha_v(t), & v(0) = v_0, \\
\omega(t+1) = \omega(t) + \alpha_\omega(t), & \omega(0) = \omega_0, \\
\boldsymbol{x}^i(t) = \pi\left(\exp(\widehat{\Omega}(t))\boldsymbol{x}_0^i(t)\lambda^i(t) + T(t)\right) + \boldsymbol{n}^i(t), & i = 1, 2, \ldots, N.
\end{cases}
$$

$\log_{SO(3)}(R)$ stands for Ω such that $R = e^{\widehat{\Omega}}$

Minimal realization

$$\begin{cases} \boldsymbol{x}_0^i(t+1) = \boldsymbol{x}_0^i(t), \quad i = \mathbf{4}, 5, \ldots, N, & \boldsymbol{x}_0^i(0) = \boldsymbol{x}_0^i, \\ \lambda^i(t+1) = \lambda^i(t), \quad i = \mathbf{2}, 3, \ldots, N, & \lambda^i(0) = \lambda_0^i, \\ T(t+1) = \exp(\widehat{\omega}(t))T(t) + v(t), & T(0) = T_0, \\ \Omega(t+1) = \log_{SO(3)}\big(\exp(\widehat{\omega}(t))\exp(\widehat{\Omega}(t))\big), & \Omega(0) = \Omega_0, \\ v(t+1) = v(t) + \alpha_v(t), & v(0) = v_0, \\ \omega(t+1) = \omega(t) + \alpha_\omega(t), & \omega(0) = \omega_0, \\ \boldsymbol{x}^i(t) = \pi\left(\exp(\widehat{\Omega}(t))\boldsymbol{x}_0^i(t)\lambda^i(t) + T(t)\right) + \boldsymbol{n}^i(t), & i = 1, 2, \ldots, N. \end{cases}$$

$$x(t) \doteq \left[\boldsymbol{x}_0^4(t)^T, \ldots, \boldsymbol{x}_0^N(t)^T, \ \lambda^2(t), \ldots, \lambda^N(t), T^T(t), \Omega^T(t), v^T(t), \omega^T(t)\right]^T,$$
$$y(t) \doteq \left[\boldsymbol{x}^1(t)^T, \ldots, \boldsymbol{x}^N(t)^T\right]^T.$$

- Now it looks very much like: $\begin{cases} x(t+1) = f(x(t)), & x(t_0) = x_0 \\ y(t) = h(x(t)), \end{cases}$

- And we are looking for (a point statistic of): $p(x(t)|y^t)$

$$\hat{x}(t|t) = E_p[x(t)] = \int x(t) dP(x(t)|y^t)$$

EFK vs. particle filter?

- For single rigid body/static scene, expect unimodal posterior
- No need to estimate entire density; point estimate suffices
- Robust (M-) version of EKF works well in practice …
- … and in real time for a few hundred feature points

In practice …

- Adding/removing features (subfilters)
- Multiple motions/outliers (M-filter, innovation tests)
- Tracking drift (reset with wide-baseline matching)
- Switching the reference features (hard! Causes unavoidable global drift)
- Global registration (maintain DB of lost features)

QuickTime™ and a
MS-MPEG4v2 Video decompressor
are needed to see this picture.