Tracking Deformable Objects with Point Clouds

John Schulman, Alex Lee, Jonathan Ho, Pieter Abbeel

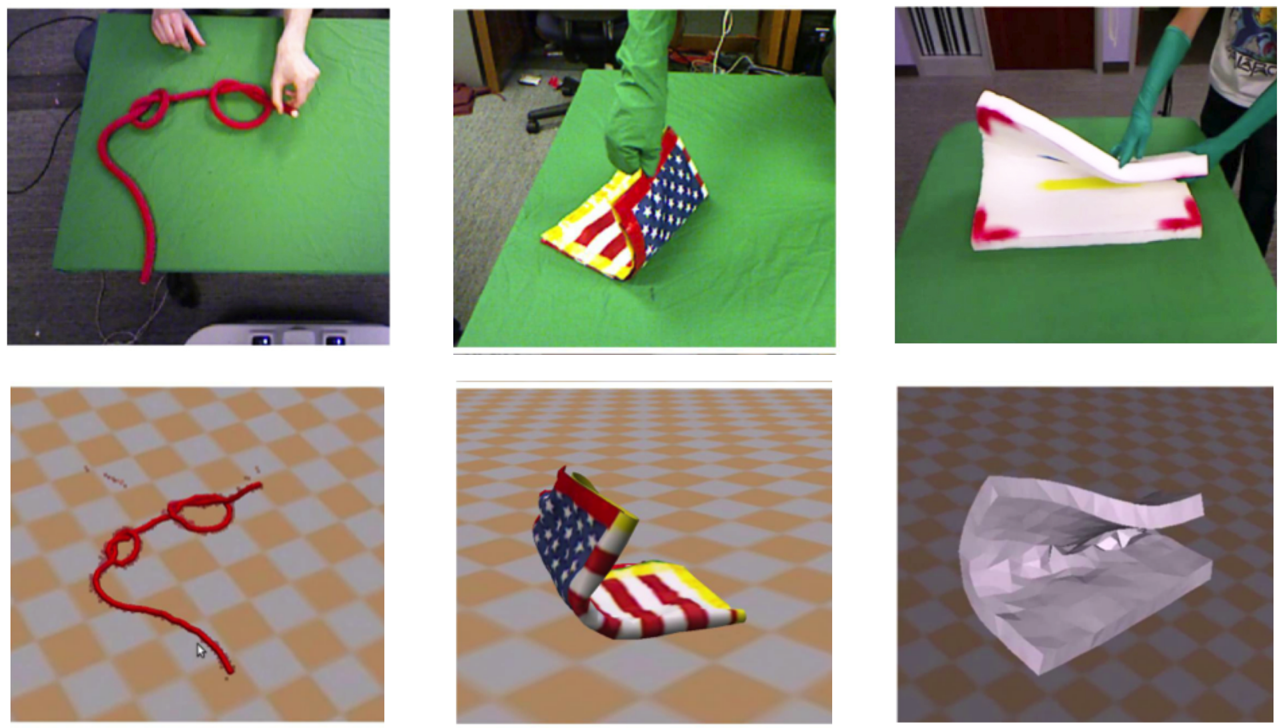
UC Berkeley, EECS Department

Thursday, July 4, 13

Hi. My name is John Schulman, and this presentation is based on joint work with Alex Lee, Jonathan Ho, and Pieter Abbeel.

Goal

- Track deformable objects from point cloud data
- Assumption: we have a physical model of the object



Kinect RGB

Rendering of state estimate

1D 2D 3D

In this work, we address the problem of continuously estimating the state of a deformable object

This is a hard problem because the object may be mostly occluded by itself or the arms of a human or robot that's manipulating it.

We can deal with these ambiguities by having a good dynamics model and ensuring that the object follows a physically plausible trajectory.

The images here show some of the objects we can track with our method--the top row shows the RGB images from the Kinect, and the bottom shows a rendering of the state estimate from our tracking algorithm.

Energy Minimization Methods

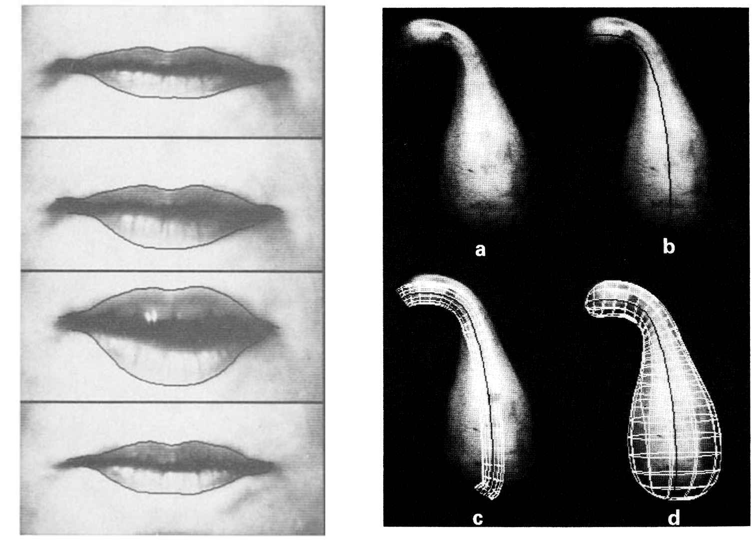
x: state estimate **y**: observation

$$E_{\text{total}}(\mathbf{x}, \mathbf{y}) = E_{\text{internal}}(\mathbf{x}) + E_{\text{external}}(\mathbf{x}, \mathbf{y})$$

discourage bending and stretching (internal)

encourage model to match up with image (external)

$$\min_{\mathbf{x}} E_{\text{total}}(\mathbf{x}, \mathbf{y})$$



Kass, Terzopoulos, Witkin, 1988

Thursday, July 4, 13

Now I'm going to give a brief background on some of the methods we build on, starting with energy minimization methods, which were pioneered by Kass, Terz, and Witkin in the late 80s.

From now on, x is going to denote our state estimate, configuration of a physical model, and y will denote the image or point cloud data.

One defines an internal energy, which discourages bending and stretching of the model, and an external energy, which encourages the model to match up with the image.

People have extended these energy minimization ideas in lots of different directions, exploring different physical models, different methods for finding correspondences, and different optimization schemes.

Energy Minimization Methods

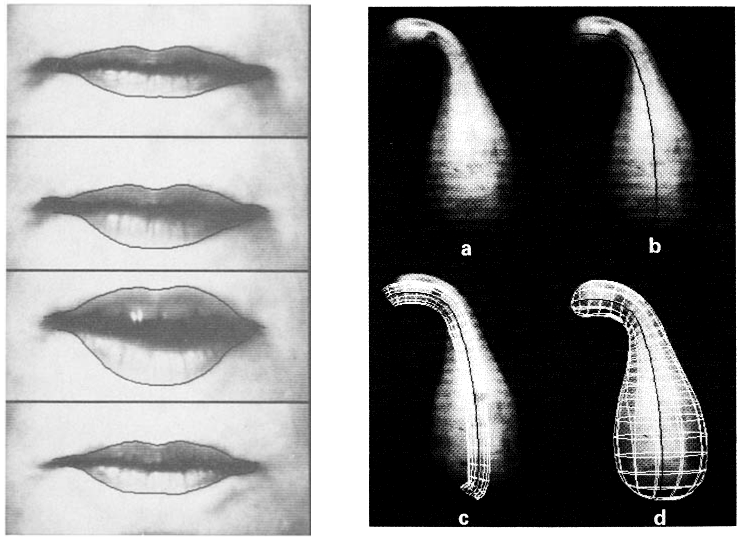
x: state estimate   **y**: observation

$$E_{\text{total}}(\mathbf{x}, \mathbf{y}) = E_{\text{internal}}(\mathbf{x}) + E_{\text{external}}(\mathbf{x}, \mathbf{y})$$

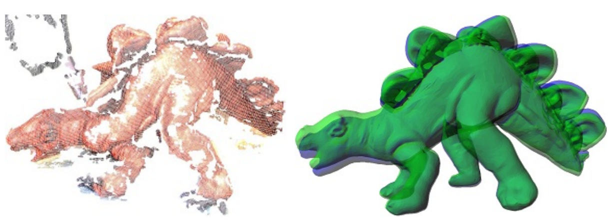
discourage bending and stretching

encourage model to match up with image

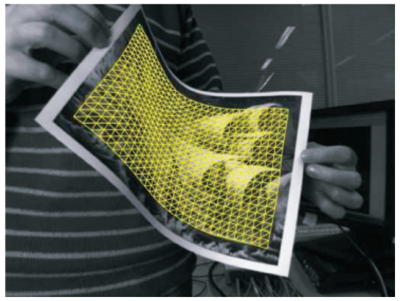
$$\min_{\mathbf{x}} E_{\text{total}}(\mathbf{x}, \mathbf{y})$$



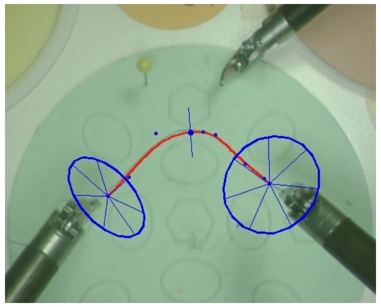
Kass, Terzopoulos, Witkin, 1988



Wuhrer, Lang, & Shu 2012



Saltzman et al. 2007



Padoy & Hager 2011

Thursday, July 4, 13

Now I'm going to give a brief background on some of the methods we build on, starting with energy minimization methods, which were pioneered by Kass, Terz, and Witkin in the late 80s.

From now on, x is going to denote our state estimate, configuration of a physical model, and y will denote the image or point cloud data.

One defines an internal energy, which discourages bending and stretching of the model, and an external energy, which encourages the model to match up with the image.

People have extended these energy minimization ideas in lots of different directions, exploring different physical models, different methods for finding correspondences, and different optimization schemes.

Probabilistic Methods

Thursday, July 4, 13

for reference i'm putting energy minimization methods up on the right here

in their simplest form, the probabilistic methods are equivalent

rather than minimizing energy, you maximize e to the negative energy

But in addition, we can now define rich models of noise and uncertainty, in particular, we can introduce unobserved variables z for correspondences--meaning, what parts of the object correspond to what observations

The three references on the bottom of the slide are some nice examples of using probabilistic models for solving registration problems involving deformable objects.

Probabilistic Methods

Energy minimization methods

$$E_{\text{total}}(\mathbf{x}, \mathbf{y}) = E_{\text{internal}}(\mathbf{x}) + E_{\text{external}}(\mathbf{x}, \mathbf{y})$$

$$\min_{\mathbf{x}} E_{\text{total}}(\mathbf{x}, \mathbf{y})$$

Thursday, July 4, 13

for reference i'm putting energy minimization methods up on the right here

in their simplest form, the probabilistic methods are equivalent

rather than minimizing energy, you maximize e to the negative energy

But in addition, we can now define rich models of noise and uncertainty, in particular, we can introduce unobserved variables z for correspondences--meaning, what parts of the object correspond to what observations

The three references on the bottom of the slide are some nice examples of using probabilistic models for solving registration problems involving deformable objects.

Probabilistic Methods

x: state estimate **y**: observation

$$p(\mathbf{x}, \mathbf{y}) \propto e^{-E_{\text{internal}}(\mathbf{x})} e^{-E_{\text{external}}(\mathbf{x},\mathbf{y})}$$

$$\max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y})$$ (MAP estimation)

Energy minimization methods

$$E_{\text{total}}(\mathbf{x}, \mathbf{y}) = E_{\text{internal}}(\mathbf{x}) + E_{\text{external}}(\mathbf{x}, \mathbf{y})$$

$$\min_{\mathbf{x}} E_{\text{total}}(\mathbf{x}, \mathbf{y})$$

Thursday, July 4, 13

for reference i'm putting energy minimization methods up on the right here

in their simplest form, the probabilistic methods are equivalent

rather than minimizing energy, you maximize e to the negative energy

But in addition, we can now define rich models of noise and uncertainty, in particular, we can introduce unobserved variables z for correspondences--meaning, what parts of the object correspond to what observations

The three references on the bottom of the slide are some nice examples of using probabilistic models for solving registration problems involving deformable objects.

Probabilistic Methods

x: state estimate **y**: observation

$$p(\mathbf{x}, \mathbf{y}) \propto e^{-E_{\text{internal}}(\mathbf{x})} e^{-E_{\text{external}}(\mathbf{x}, \mathbf{y})}$$

$$\max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y})$$ (MAP estimation)

Energy minimization methods

$$E_{\text{total}}(\mathbf{x}, \mathbf{y}) = E_{\text{internal}}(\mathbf{x}) + E_{\text{external}}(\mathbf{x}, \mathbf{y})$$

$$\min_{\mathbf{x}} E_{\text{total}}(\mathbf{x}, \mathbf{y})$$

$$p(\mathbf{x}, \mathbf{y}) \propto \sum_{\mathbf{z}} e^{-E(\mathbf{x}, \mathbf{y}, \mathbf{z})}$$

z: correspondences

Thursday, July 4, 13

for reference i'm putting energy minimization methods up on the right here

in their simplest form, the probabilistic methods are equivalent

rather than minimizing energy, you maximize e to the negative energy

But in addition, we can now define rich models of noise and uncertainty, in particular, we can introduce unobserved variables z for correspondences--meaning, what parts of the object correspond to what observations

The three references on the bottom of the slide are some nice examples of using probabilistic models for solving registration problems involving deformable objects.

Probabilistic Methods

x: state estimate **y**: observation

$$p(\mathbf{x},\mathbf{y}) \propto e^{-E_{\text{internal}}(\mathbf{x})} e^{-E_{\text{external}}(\mathbf{x},\mathbf{y})}$$

$$\max_{\mathbf{x}} p(\mathbf{x},\mathbf{y})$$ (MAP estimation)

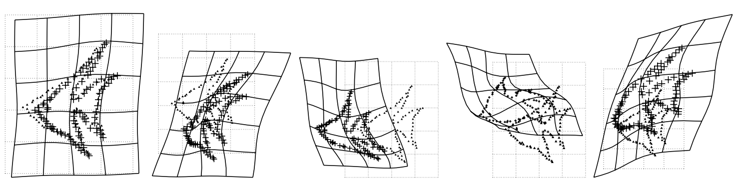
Energy minimization methods

$$E_{\text{total}}(\mathbf{x},\mathbf{y}) = E_{\text{internal}}(\mathbf{x}) + E_{\text{external}}(\mathbf{x},\mathbf{y})$$

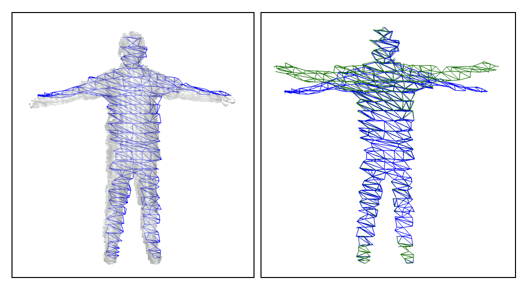
$$\min_{\mathbf{x}} E_{\text{total}}(\mathbf{x},\mathbf{y})$$

$$p(\mathbf{x},\mathbf{y}) \propto \sum_{\mathbf{z}} e^{-E(\mathbf{x},\mathbf{y},\mathbf{z})}$$

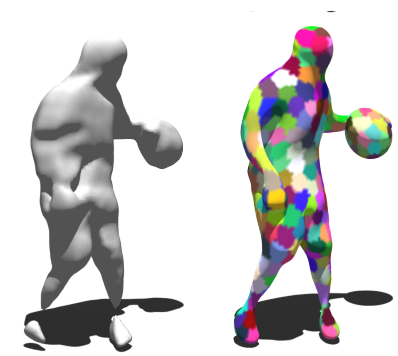
z: correspondences



Myronenko & Song 2007



Hahnel, Thrun & Burgard 2003



Cagniart, Boyer, & Ilic 2010

Thursday, July 4, 13

for reference i'm putting energy minimization methods up on the right here

in their simplest form, the probabilistic methods are equivalent

rather than minimizing energy, you maximize e to the negative energy

But in addition, we can now define rich models of noise and uncertainty, in particular, we can introduce unobserved variables z for correspondences--meaning, what parts of the object correspond to what observations

The three references on the bottom of the slide are some nice examples of using probabilistic models for solving registration problems involving deformable objects.

Challenges

Thursday, July 4, 13

To summarize the key challenges, on the observation modeling side, we need to acount for unknown correspondences, noise and occlusions
on the other hand, we have all these physical constraints to enforce, for example, no collisions, no stretching

Probabilistic models have excelled at the observation models, and the energy based methods have excelled at enforcing physical constraints.
But really we'd like to get the best of both worlds.

The the approach I'll present will address all of these challenges through the following contributions:
It uses a probabilistic model that captures all of these factors
[read slide]

Challenges

- Observation modeling:
  - correspondence problem
  - noise
  - occlusions

To summarize the key challenges, on the observation modeling side, we need to acount for unknown correspondences, noise and occlusions
on the other hand, we have all these physical constraints to enforce, for example, no collisions, no stretching

Probabilistic models have excelled at the observation models, and the energy based methods have excelled at enforcing physical constraints.
But really we'd like to get the best of both worlds.

The the approach I'll present will address all of these challenges through the following contributions:
It uses a probabilistic model that captures all of these factors
[read slide]

Challenges

- Observation modeling:
  - correspondence problem
  - noise
  - occlusions
- Physical constraints:
  - non-penetration
  - hard constraints on bending and stretching

Thursday, July 4, 13

To summarize the key challenges, on the observation modeling side, we need to acount for unknown correspondences, noise and occlusions
on the other hand, we have all these physical constraints to enforce, for example, no collisions, no stretching

Probabilistic models have excelled at the observation models, and the energy based methods have excelled at enforcing physical constraints.
But really we'd like to get the best of both worlds.

The the approach I'll present will address all of these challenges through the following contributions:
It uses a probabilistic model that captures all of these factors
[read slide]

Challenges Contributions

- Observation modeling:
  - correspondence problem
  - noise
  - occlusions
- Physical constraints:
  - non-penetration
  - hard constraints on bending and stretching

Thursday, July 4, 13

To summarize the key challenges, on the observation modeling side, we need to acount for unknown correspondences, noise and occlusions
on the other hand, we have all these physical constraints to enforce, for example, no collisions, no stretching

Probabilistic models have excelled at the observation models, and the energy based methods have excelled at enforcing physical constraints.
But really we'd like to get the best of both worlds.

The the approach I'll present will address all of these challenges through the following contributions:
It uses a probabilistic model that captures all of these factors
[read slide]

Challenges Contributions

- Observation modeling:
  - correspondence problem
  - noise
  - occlusions
- Physical constraints:
  - non-penetration
  - hard constraints on bending and stretching

- Modeling contribution:
  - Probabilistic model that captures correspondence problem, noise, and occlusions

To summarize the key challenges, on the observation modeling side, we need to acount for unknown correspondences, noise and occlusions
on the other hand, we have all these physical constraints to enforce, for example, no collisions, no stretching

Probabilistic models have excelled at the observation models, and the energy based methods have excelled at enforcing physical constraints.
But really we'd like to get the best of both worlds.

The the approach I'll present will address all of these challenges through the following contributions:
It uses a probabilistic model that captures all of these factors
[read slide]

Challenges Contributions

- Observation modeling:
  - correspondence problem
  - noise
  - occlusions
- Physical constraints:
  - non-penetration
  - hard constraints on bending and stretching

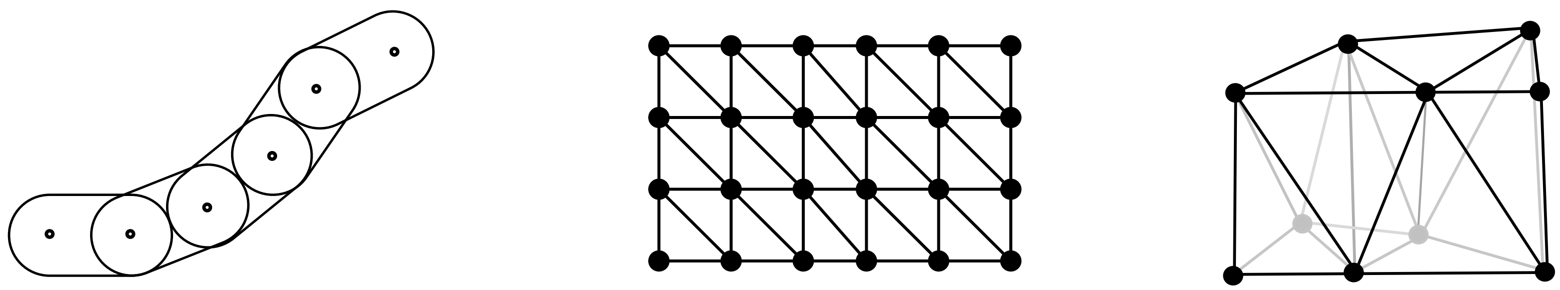
- Modeling contribution:
  - Probabilistic model that captures correspondence problem, noise, and occlusions
- Algorithmic contribution:
  - Modification of the EM algorithm that accounts for physical constraints
  - Operates by only introducing external forces into physics simulation engines

To summarize the key challenges, on the observation modeling side, we need to acount for unknown correspondences, noise and occlusions
on the other hand, we have all these physical constraints to enforce, for example, no collisions, no stretching

Probabilistic models have excelled at the observation models, and the energy based methods have excelled at enforcing physical constraints.
But really we'd like to get the best of both worlds.

The the approach I'll present will address all of these challenges through the following contributions:
It uses a probabilistic model that captures all of these factors
[read slide]

Preliminaries



Thursday, July 4, 13

Our approach can work with a wide ragne of physical models-1d,2d,3d models

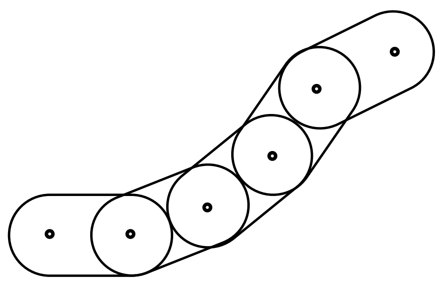
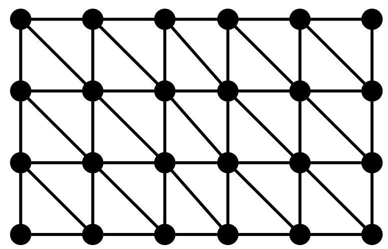
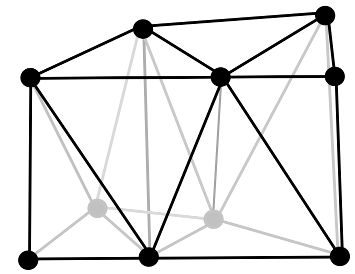
the algorithm assumes some image and point cloud processing has happened, which distinguishes between object and bacground

but it’s not assumed to be perfect

here’s a typical example of what comes out of the preprocessing

Preliminaries

- Physical models

Thursday, July 4, 13

Our approach can work with a wide ragne of physical models-1d,2d,3d models

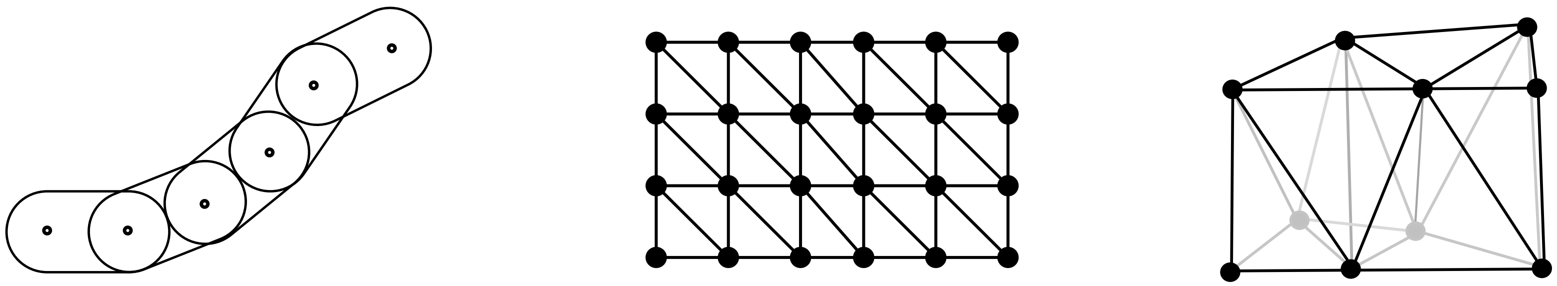
the algorithm assumes some image and point cloud processing has happened, which distinguishes between object and bacground

but it’s not assumed to be perfect

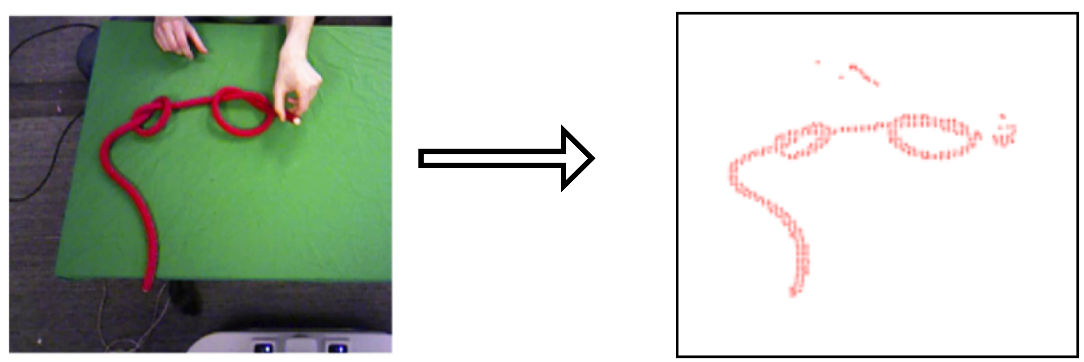
here’s a typical example of what comes out of the preprocessing

Preliminaries

- Physical models



- Image / point cloud processing
  - background subtraction (may have false positives & negatives)



Thursday, July 4, 13

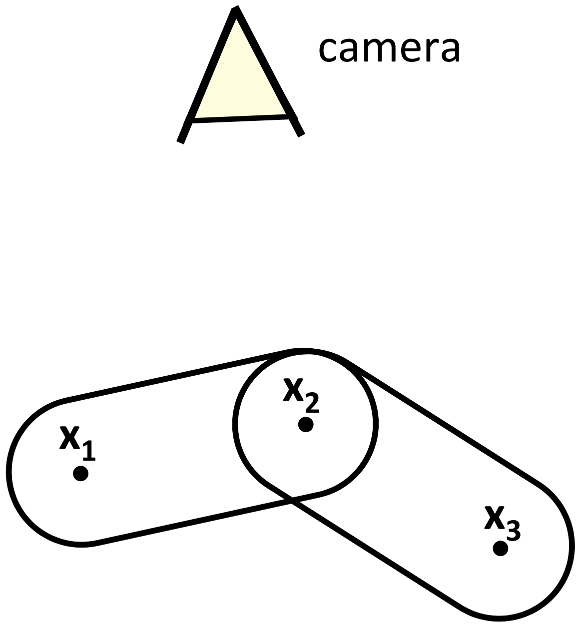
Our approach can work with a wide ragne of physical models-1d,2d,3d models

the algorithm assumes some image and point cloud processing has happened, which distinguishes between object and bacground

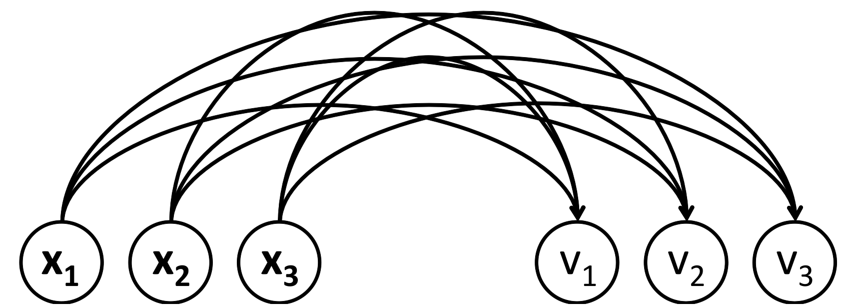
but it’s not assumed to be perfect

here’s a typical example of what comes out of the preprocessing

Observation Model



Graphical model



Thursday, July 4, 13

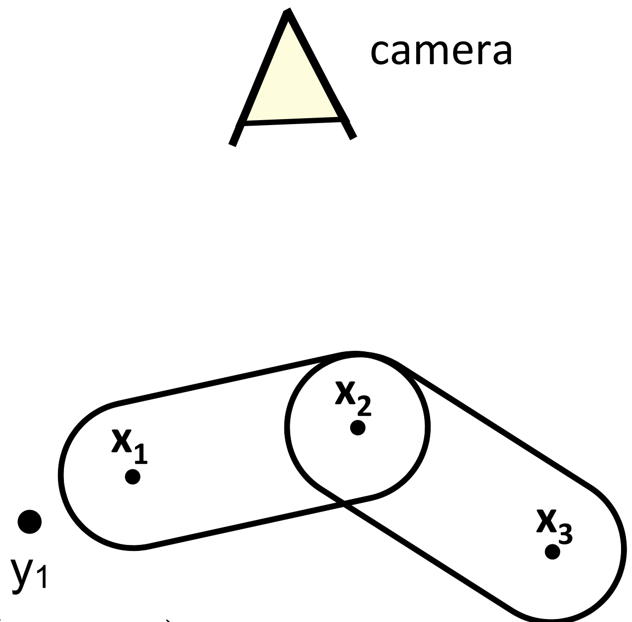
Now I'll describe the probabilistic generative model for the observations. Let's assume this here is the state of the object, and let's see how we can generate an observation.

First we need to see which parts are visible, and we have these variables v1,v2,v3 indicating whether x1,x2,x3 are visible.

Then we generate variable z1 which tells us for our first observation which one of x1,x2,x3 is going to be observed
All of x1,x2,x3 are visible, so it could pick any one of them, and in this example it happened to pick x1 so it's 1,0,0
the sensor is noisy, so we get x1 + some gaussian noise, which results in our first measurement y1

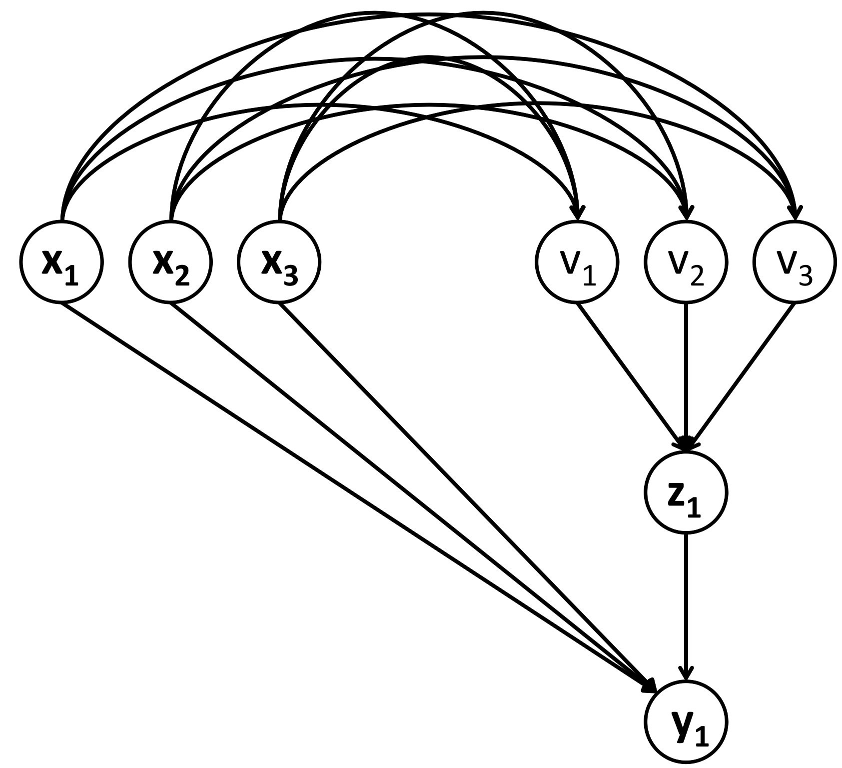
this process is now repeated

Observation Model



$$z_1 = (1, 0, 0)$$
$$y_1 = \mathcal{N}(\mathbf{x}_1, \sigma^2)$$

Graphical model



Thursday, July 4, 13

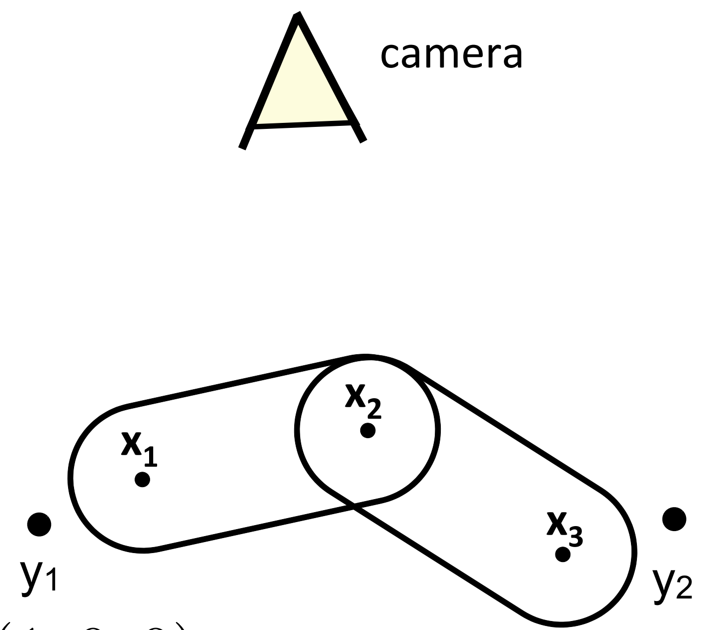
Now I'll describe the probabilistic generative model for the observations. Let's assume this here is the state of the object, and let's see how we can generate an observation.

First we need to see which parts are visible, and we have these variables v1,v2,v3 indicating whether x1,x2,x3 are visible.

Then we generate variable z1 which tells us for our first observation which one of x1,x2,x3 is going to be observed
All of x1,x2,x3 are visible, so it could pick any one of them, and in this example it happened to pick x1 so it's 1,0,0
the sensor is noisy, so we get x1 + some gaussian noise, which results in our first measurement y1

this process is now repeated

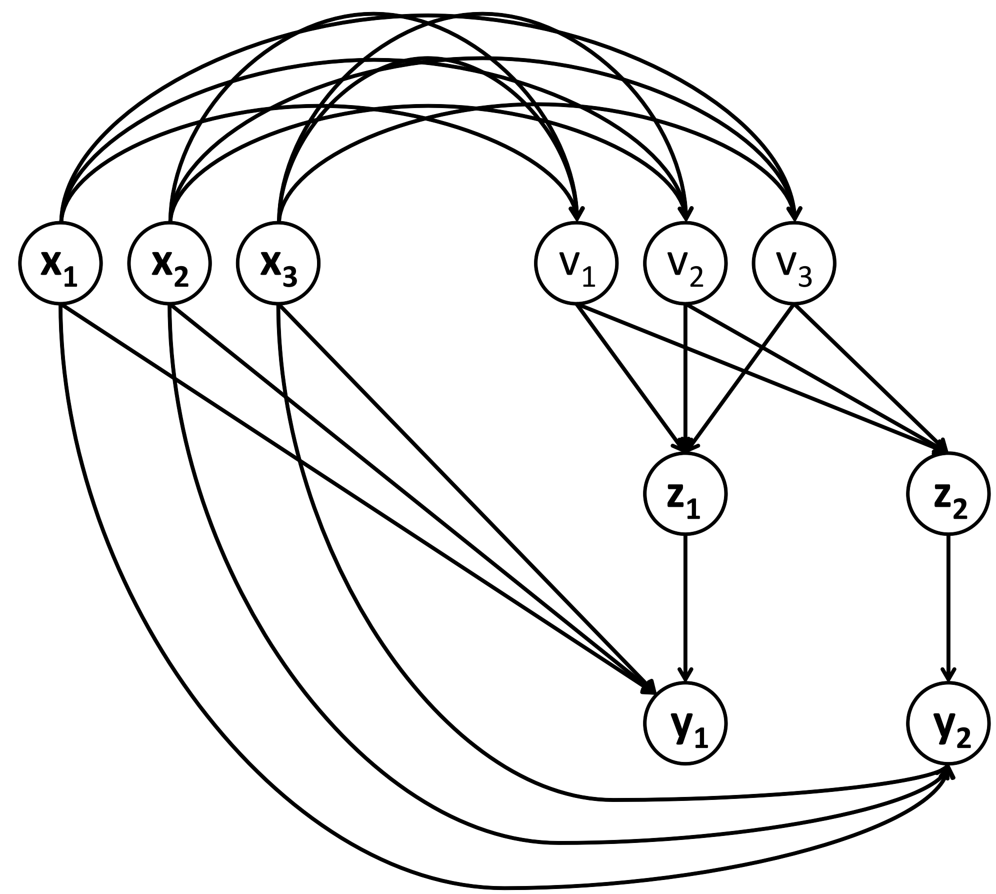
Observation Model



$$z_1 = (1, 0, 0)$$
$$y_1 = \mathcal{N}(\mathbf{x}_1, \sigma^2)$$

$$z_2 = (0, 0, 1)$$
$$y_2 = \mathcal{N}(\mathbf{x}_3, \sigma^2)$$

Graphical model



Thursday, July 4, 13

Now I'll describe the probabilistic generative model for the observations. Let's assume this here is the state of the object, and let's see how we can generate an observation.

First we need to see which parts are visible, and we have these variables v1,v2,v3 indicating whether x1,x2,x3 are visible.

Then we generate variable z1 which tells us for our first observation which one of x1,x2,x3 is going to be observed
All of x1,x2,x3 are visible, so it could pick any one of them, and in this example it happened to pick x1 so it's 1,0,0
the sensor is noisy, so we get x1 + some gaussian noise, which results in our first measurement y1

this process is now repeated

EM Algorithm

Thursday, July 4, 13

Here’s the problem we’re trying to solve--we’re trying to find the most probable state x, and we’ve marginalized out the latent variables z

The standard way to solve problems like this is to use the em algorithm:

you alternate between the e step, where you calculate the posterior distribution of the latent variables z

and the m step, which uses the calculated distribution over z to update x

that doesn’t work for the tracking problem because that would violate the physical constraints.

so let’s reexamine the m step and see what we can do.

EM Algorithm

- MAP inference problem:

$$\arg\max_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{x}} \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

Here's the problem we're trying to solve--we're trying to find the most probable state x, and we've marginalized out the latent variables z

The standard way to solve problems like this is to use the em algorithm:

you alternate between the e step, where you calculate the posterior distribution of the latent variables z

and the m step, which uses the calculated distribution over z to update x

that doesn't work for the tracking problem because that would violate the physical constraints.

so let's reexamine the m step and see what we can do.

EM Algorithm

- MAP inference problem:

$$\arg\max_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{x}} \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

- For i=1,2,3,...

Thursday, July 4, 13

Here’s the problem we’re trying to solve--we’re trying to find the most probable state x, and we’ve marginalized out the latent variables z

The standard way to solve problems like this is to use the em algorithm:

you alternate between the e step, where you calculate the posterior distribution of the latent variables z

and the m step, which uses the calculated distribution over z to update x

that doesn’t work for the tracking problem because that would violate the physical constraints.

so let’s reexamine the m step and see what we can do.

EM Algorithm

- MAP inference problem:

$$\arg\max_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{x}} \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

- For i=1,2,3,...

E Step: $q^{(i)}(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}^{(i-1)}, \mathbf{y})$

Calculate posterior of latent variables

Thursday, July 4, 13

Here's the problem we're trying to solve--we're trying to find the most probable state x, and we've marginalized out the latent variables z
The standard way to solve problems like this is to use the em algorithm:
you alternate between the e step, where you calculate the posterior distribution of the latent variables z
and the m step, which uses the calculated distribution over z to update x
that doesn't work for the tracking problem because that would violate the physical constraints.
so let's reexamine the m step and see what we can do.

EM Algorithm

- MAP inference problem:

$$\arg\max_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{x}} \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

- For i=1,2,3,...

E Step: $$q^{(i)}(\mathbf{z}) = p(\mathbf{z} | \mathbf{x}^{(i-1)}, \mathbf{y})$$

Calculate posterior of latent variables

M Step: $$\mathbf{x}^{(i)} = \arg\max_{\mathbf{x}} A_{q^{(i)}}(\mathbf{x})$$

Maximize a lower bound to log-probability

$$A_{q^{(i)}}(\mathbf{x}) = \sum_{\mathbf{z}} q^{(i)}(\mathbf{z}) \log p(\mathbf{x}, \mathbf{y}, \mathbf{z}) + S(q^{(i)}) \leq \log p(\mathbf{x}, \mathbf{y})$$

Thursday, July 4, 13

Here’s the problem we’re trying to solve--we’re trying to find the most probable state x, and we’ve marginalized out the latent variables z
The standard way to solve problems like this is to use the em algorithm:
you alternate between the e step, where you calculate the posterior distribution of the latent variables z
and the m step, which uses the calculated distribution over z to update x
that doesn’t work for the tracking problem because that would violate the physical constraints.
so let’s reexamine the m step and see what we can do.

Modified M Step

Thursday, July 4, 13

For your reference, here's the standard M step

The key insight we use is can do this constrained minimization by repeatedly applying forces to our object model in simulation, where the forces are based on gradient of the objective.

if you move to the maximum this way, you can enforce the physical constraints in your optimization

The steps taken by this minimization procedure have a physical interpretation: these are the dynamics you obtain if you have a physical system where the -A is the potential energy, and you do simulation steps.

And a damped physical system will converge to a local minimum of the potential energy, so this procedure converges to the local maximum of A.

Modified M Step

- Standard M step (i^{th} iteration)

$$\mathbf{x}^{(i)} = \arg\max_{\mathbf{x}} A_{q^{(i)}}(\mathbf{x})$$

For your reference, here's the standard M step

The key insight we use is can do this constrained minimization by repeatedly applying forces to our object model in simulation, where the forces are based on gradient of the objective.

if you move to the maximum this way, you can enforce the physical constraints in your optimization

The steps taken by this minimization procedure have a physical interpretation: these are the dynamics you obtain if you have a physical system where the -A is the potential energy, and you do simulation steps.

And a damped physical system will converge to a local minimum of the potential energy, so this procedure converges to the local maximum of A.

Modified M Step

- Standard M step (i^th iteration)

$$\mathbf{x}^{(i)} = \arg\max_{\mathbf{x}} A_{q^{(i)}}(\mathbf{x})$$

- M step with physical constraints:
  - Repeatedly apply forces in gradient direction (+ damping) until convergence

$$\mathbf{f}_k = \nabla_{\mathbf{x}_k} A_q(\mathbf{x}) - \beta\dot{\mathbf{x}}$$

For your reference, here's the standard M step

The key insight we use is can do this constrained minimization by repeatedly applying forces to our object model in simulation, where the forces are based on gradient of the objective.

if you move to the maximum this way, you can enforce the physical constraints in your optimization

The steps taken by this minimization procedure have a physical interpretation: these are the dynamics you obtain if you have a physical system where the -A is the potential energy, and you do simulation steps.

And a damped physical system will converge to a local minimum of the potential energy, so this procedure converges to the local maximum of A.

Modified M Step

- Standard M step (i[th] iteration)

$$\mathbf{x}^{(i)} = \arg\max_{\mathbf{x}} A_{q^{(i)}}(\mathbf{x})$$

- M step with physical constraints:
  - Repeatedly apply forces in gradient direction (+ damping) until convergence

$$\mathbf{f}_k = \nabla_{\mathbf{x}_k} A_q(\mathbf{x}) - \beta \dot{\mathbf{x}}$$

  - physical interpretation:
    - simulating a system where $-A_{q^{(i)}}(\mathbf{x})$ is the potential energy,
    - damped physical system converges to local minimum of potential energy

For your reference, here's the standard M step

The key insight we use is can do this constrained minimization by repeatedly applying forces to our object model in simulation, where the forces are based on gradient of the objective.

if you move to the maximum this way, you can enforce the physical constraints in your optimization

The steps taken by this minimization procedure have a physical interpretation: these are the dynamics you obtain if you have a physical system where the -A is the potential energy, and you do simulation steps.

And a damped physical system will converge to a local minimum of the potential energy, so this procedure converges to the local maximum of A.

Real-time Implementation

Thursday, July 4, 13

In the real time setting, we just do as many EM iterations as possible after each new point cloud is received

----

That graphical model and iference procedure describes what to do for a single image received, assuming that you have an initial estimate of the state--
you do the EM algoirhm until convergence based on that new image.

Real-time Implementation

- For time t=1,2,...
  - Iterate until next point cloud received:
    - E Step
    - M Step

In the real time setting, we just do as many EM iterations as possible after each new point cloud is received

----

That graphical model and iference procedure describes what to do for a single image received, assuming that you have an initial estimate of the state--
you do the EM algoirhm until convergence based on that new image.

Demo video

Thursday, July 4, 13

Here’s a demonstration of our algorithm in action.

The algorithm runs in real time--the next two videos of rope manipulation are just screen-captures of the algorithm running on live data.

However, note that the video is sped up 4x.

Here you can see the algorithm running on rope during knot tying.

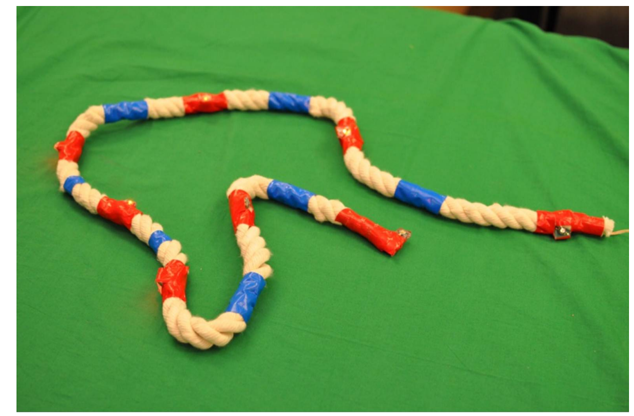
By the way, we used bullet physics engine--an open source physics engine designed from games which is also popular in robotics. The nice thing is that then we can just throw in a robot. now the algorithm knows about the occlusions from the robot, because it’s doing raycast, and it knows about the collision and physical interactions between the robot and rope, which greatly improves the tracking.

The next two videos of cloth were run as part of our ground-truth data collection setup. So they have markers on them and lots of texture because we wanted to see how well it worked. We actually are using the color here, in a pretty straightforward extension of algorithm I just described to include color--the color makes it work a bit more accurately, though it works without color too. We’re not using all the textures information though, just the LAB color values.

Quantitative Evaluation of Accuracy

PHASESPACE

Motion capture



Thursday, July 4, 13

We also performed quantitative experiments, which enabled us to evaluate the robustness and accuracy of our algorithm.

we lined a rope and a flag with LED markers from a commercial motion capture system and recorded videos of them being manipulated by a human or robot, along with the ground truth data of the location of all visible markers

Experimental results

Thursday, July 4, 13

in total, we had 14 video sequences.

We tested them by running the algorithm with the same parameters, of course.

There were a few failure cases, where the tracking got lost, and couldn't recover, but overall, the tracking algorithm usually managed to track with only a few centimeters mean error.

Experimental results

- 14 different tasks
  - human manipulates rope
  - robot manipulates rope
  - human manipulates cloth

in total, we had 14 video sequences.

We tested them by running the algorithm with the same parameters, of course.

There were a few failure cases, where the tracking got lost, and couldn't recover, but overall, the tracking algorithm usually managed to track with only a few centimeters mean error.

Experimental results

- 14 different tasks
  - human manipulates rope
  - robot manipulates rope
  - human manipulates cloth
- Succeeds on 13/14 tasks with 2.5cm mean error on successful runs

in total, we had 14 video sequences.

We tested them by running the algorithm with the same parameters, of course.

There were a few failure cases, where the tracking got lost, and couldn't recover, but overall, the tracking algorithm usually managed to track with only a few centimeters mean error.

Summary

Thursday, July 4, 13

to track an object through deformation

we embed it in a simulation

latent variables match targets to sources

find max of log p by applying forces

we even have a real time implementation

which works for robotic manipulation

Summary

- Modeling contribution:
  - Probabilistic model that captures correspondence problem, noise, and occlusions

Thursday, July 4, 13

to track an object through deformation
we embed it in a simulation

latent variables match targets to sources
find max of log p by applying forces

we even have a real time implementation
which works for robotic manipulation

Summary

- Modeling contribution:
  - Probabilistic model that captures correspondence problem, noise, and occlusions
- Algorithmic contribution:
  - Modification of the EM algorithm that can account for physical constraints
  - Operates by only introducing external forces into physics simulation engines
    - Use your favorite physics engine!

Thursday, July 4, 13

to track an object through deformation
we embed it in a simulation

latent variables match targets to sources
find max of log p by applying forces

we even have a real time implementation
which works for robotic manipulation

Summary

- Modeling contribution:
  - Probabilistic model that captures correspondence problem, noise, and occlusions
- Algorithmic contribution:
  - Modification of the EM algorithm that can account for physical constraints
  - Operates by only introducing external forces into physics simulation engines
    - Use your favorite physics engine!
- Implementation runs in real-time on standard computer

Thursday, July 4, 13

to track an object through deformation
we embed it in a simulation

latent variables match targets to sources
find max of log p by applying forces

we even have a real time implementation
which works for robotic manipulation

Thanks

Thursday, July 4, 13

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

Thursday, July 4, 13

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

Thursday, July 4, 13

that concludes my talk, thanks for your attention.

Thanks

- Acknowledgements
  - Financial support: Intel, AFOSR-YIP
  - Open-source software: Bullet, PCL, OpenCV, ROS, OpenRAVE
- Code, data, paper:
  - http://rll.berkeley.edu/tracking
- Contact
  - John Schulman: joschu@eecs.berkeley.edu

that concludes my talk, thanks for your attention.