Probabilistic Fitting

Marcel Lüthi,

University of Basel

Slides based on presentation by Sandro Schönborn

Outline

- Bayesian inference
- Fitting using Markov Chain Monte Carlo
- Exercise: MCMC in Scalismo
- Fitting 3D Landmarks

Bayesian inference

Probabilities: What are they?

Four possible interpretations:

1. Long-term frequencies
   - Relative frequency of an event over time
2. Physical tendencies (propensities)
   - Arguments about a physical situation (causes of relative frequencies)
3. Degree of belief (Bayesian probabilities)
   - Subjective beliefs about events/hypothesis/facts
4. (Logic)
   - Degree of logical support for a particular hypothesis

Bayesian probabilities for image analysis

- Bayesian probabilities make sense where frequentists interpretations are not applicable!
  - No amount of repetition makes image sharp.
    - Uncertainty is not due to random effect, but because of bad telescope.
  - Still possible to use Bayesian inference.
    - Uncertainty summarizes our ignorance.

Gallileo's view on Saturn

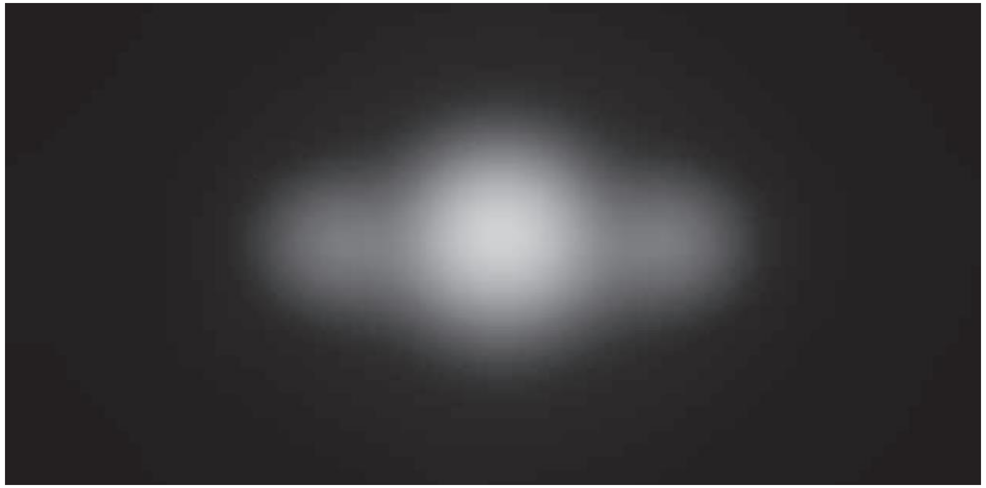


Image credit: McElrath, Statistical Rethinking: Figure 1.12

Degree of belief: An example

- Dentist example: *Does the patient have a cavity?*

$$P(\text{cavity}) = 0.1$$

$$P(\text{cavity}|\text{toothache}) = 0.8$$

$$P(\text{cavity}|\text{toothache}, \text{gum problems}) = 0.4$$

But the patient either has a cavity or does not

- *There is no 80% cavity!*
- *Having a cavity should not depend on whether the patient has a toothache or gum problems*

All these statements do not contradict each other, they summarize ***the dentist's knowledge*** about the patient

AIMA: Russell & Norvig, *Artificial Intelligence. A Modern Approach,* 3rd edition,

Uncertainty: Bayesian Probability

- Bayesian probabilities rely on a *subjective* perspective:
  - Probabilities express our *current knowledge*.
  - Can *change* when we learn or see more
  - More data -> more *certain* about our result.

Subjectivity: There is no single, real underlying distribution. A probability distribution expresses our knowledge – It is different in different situations and for different observers since they have different knowledge.

  - Subjective != Arbitrary
  - Given belief, conclusions follow by laws of probability calculus

Two important rules

Probabilistic model: joint distribution of points

$$P(x_1, x_2)$$

Marginal

Distribution of certain points only

$$P(x_1) = \sum_{x_2} P(x_1, x_2)$$

Conditional

Distribution of points conditioned on *known* values of others

$$P(x_1|x_2) = \frac{P(x_1, x_2)}{P(x_2)}$$

Product rule: $P(x_1, x_2) = p(x_1|x_2)p(x_2)$

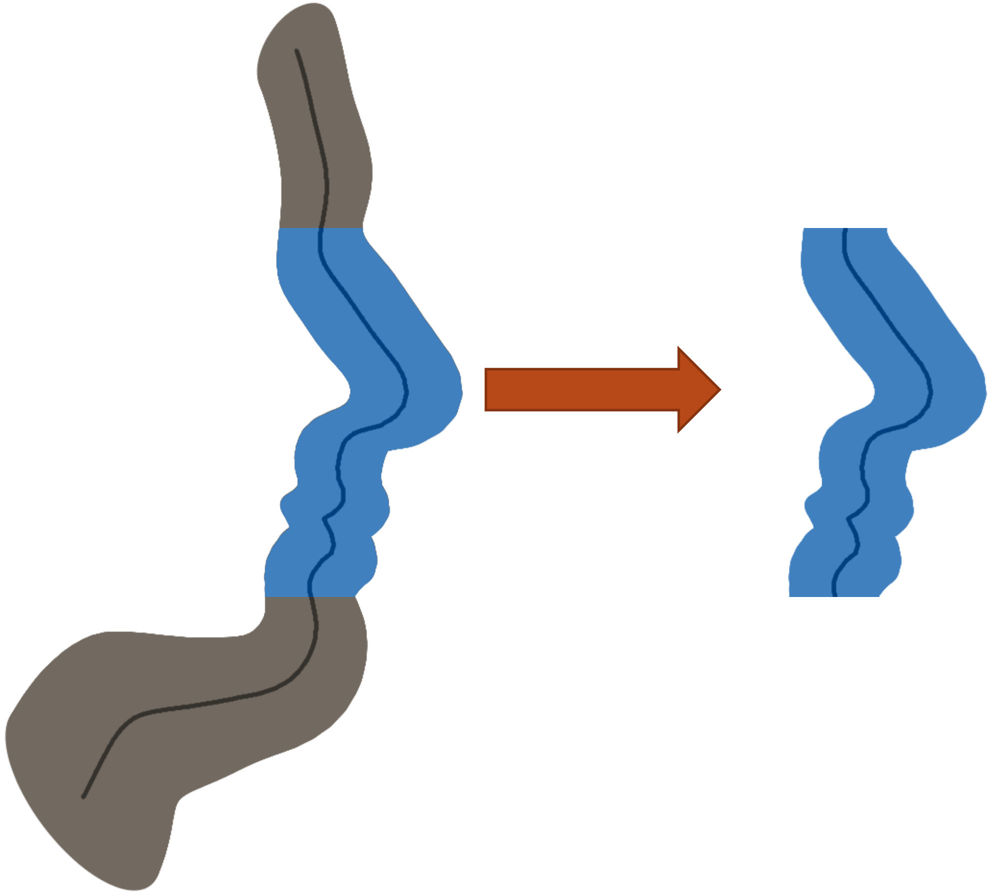
Marginalization

- Models contain irrelevant/hidden variables

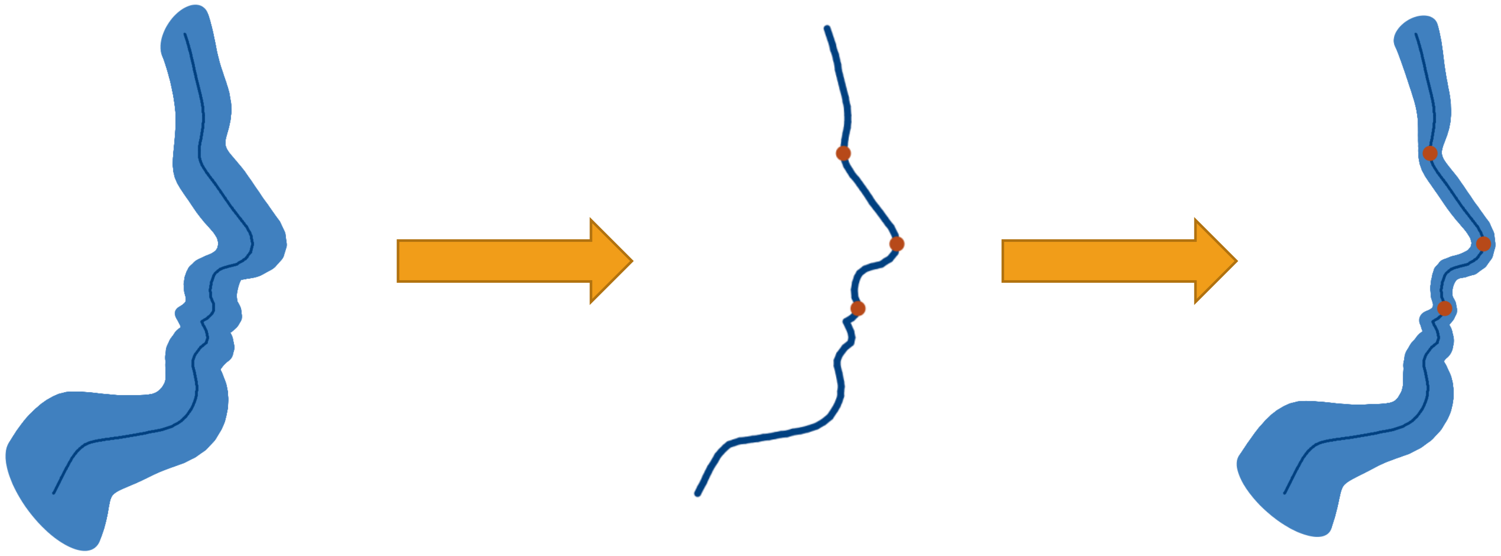
  e.g. points on chin when nose is queried

- Marginalize over hidden variables (H)

$$P(X) = \sum_H P(X, H)$$



Belief Updates



Model
Face distribution

Prior belief

Observation
Concrete points
Possibly uncertain

More knowledge

Posterior
Face distribution
consistent with observation

Posterior belief

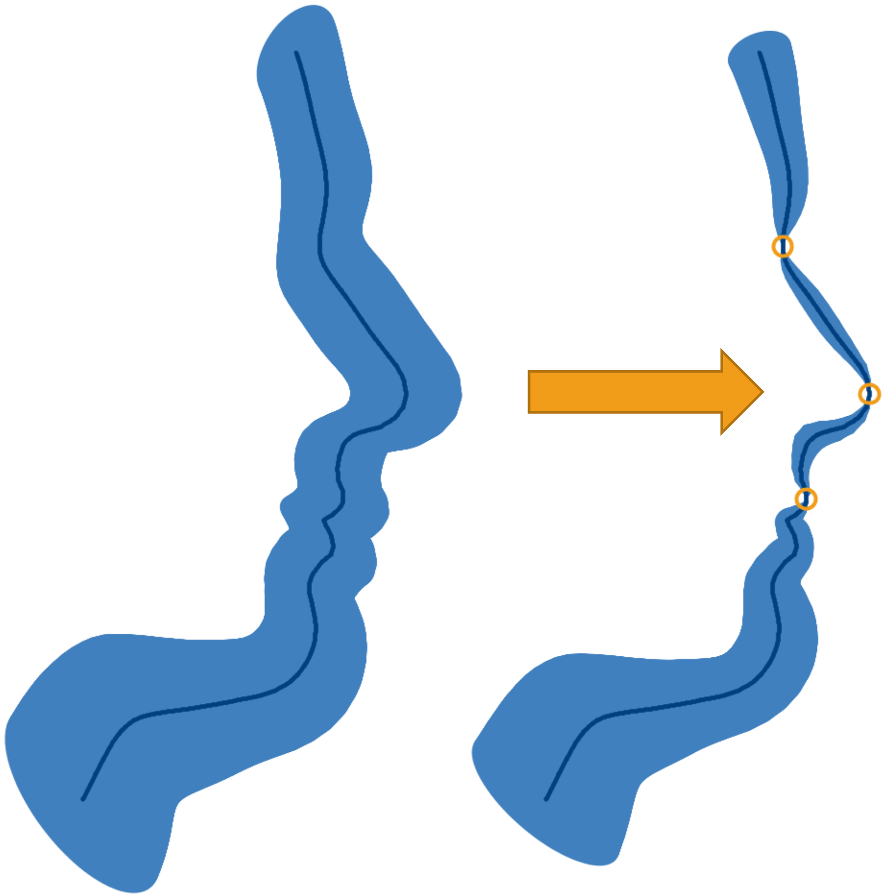
Certain Observation

- Observations are *known* values
- Distribution of X after *observing* $x_1, \dots, x_N$:

$$P(X|x_1 \dots x_N)$$

- Conditional probability

$$P(X|x_1 \dots x_N) = \frac{P(X, x_1, \dots, x_N)}{P(x_1, \dots, x_N)}$$



Towards Bayesian Inference

- Update belief about X by *observing* $x_1, \dots, x_N$

$$P(X) \to P(X|x_1 \dots x_N)$$

- Factorize joint distribution

$$P(X, x_1, \dots, x_N) = P(x_1, \dots, x_N|X)P(X)$$

- Rewrite conditional distribution

$$P(X|x_1 \dots x_N) = \frac{P(X, x_1, \dots, x_N)}{P(x_1, \dots, x_N)} = \frac{P(x_1, \dots, x_N|X)P(X)}{P(x_1, \dots, x_N)}$$

- General: Query (Q) and Evidence (E)

$$P(Q|E) = \frac{P(Q, E)}{P(E)} = \frac{P(E|Q)P(Q)}{P(E)}$$

Uncertain Observation

- Observations with *uncertainty*

  Model needs to describe how observations are distributed

  with joint distribution $P(Q, E)$

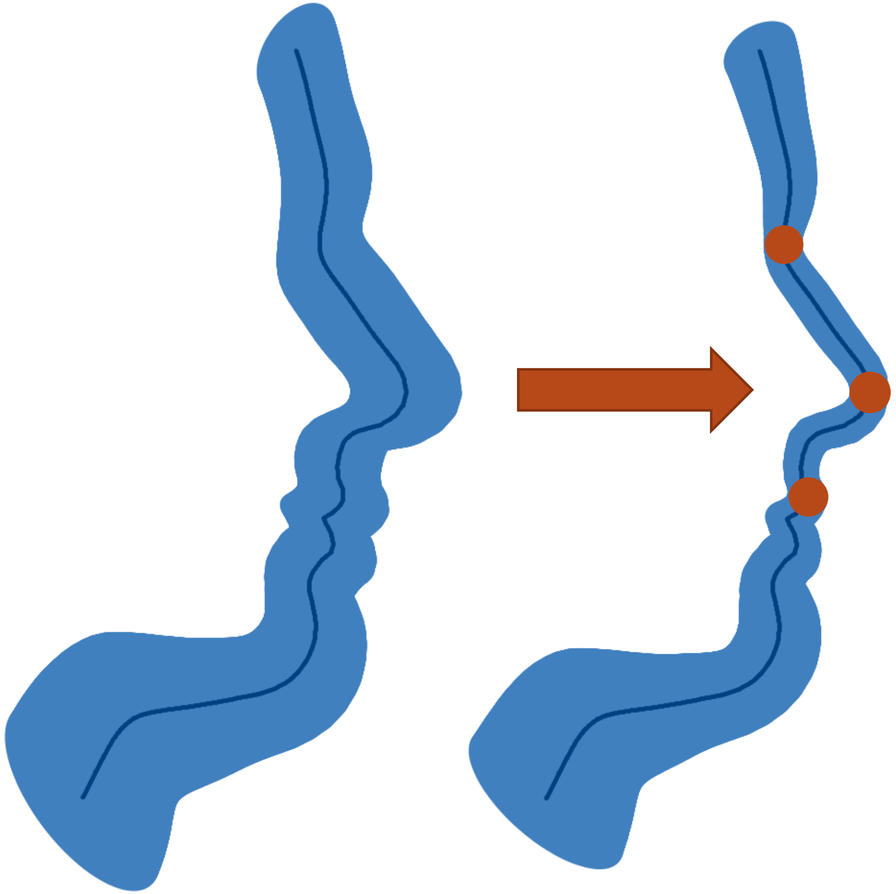
- Still conditional probability

  But joint distribution is more complex

- Joint distribution factorized

$$P(Q, E) = P(E|Q)P(Q)$$

  - Likelihood $P(E|Q)$
  - Prior $P(Q)$



Likelihood

$$\underbrace{P(Q,E)}_{\text{Joint}} = \underbrace{P(E|Q)}_{\text{Likelihood}}\underbrace{P(Q)}_{\text{Prior}}$$

- *Likelihood x prior:* factorization is more flexible than full joint
  - Prior: distribution of core model *without observation*
  - Likelihood: describes how observations are distributed

Bayesian Inference

- Conditional/Bayes rule: method to update *beliefs*

$$\underbrace{P(Q|E)}_{\text{Posterior}} = \frac{\overbrace{P(E|Q)}^{\text{Likelihood}}\overbrace{P(Q)}^{\text{Prior}}}{\underbrace{P(E)}_{\text{Marginal Likelihood}}}$$

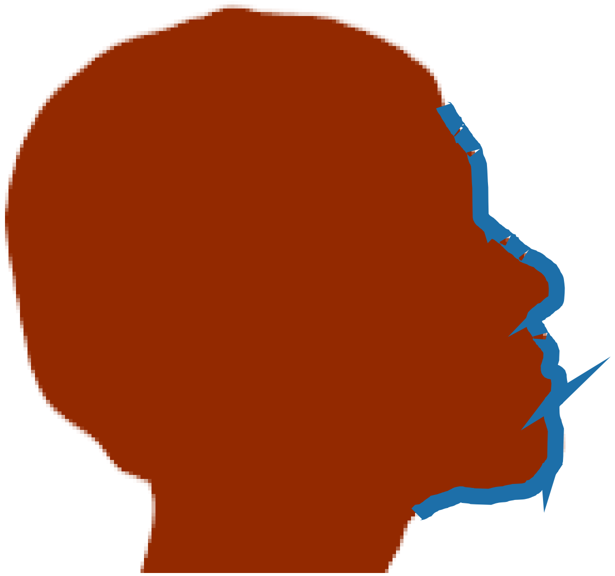
- Each observation updates our belief (changes knowledge!)

$$P(Q) \rightarrow P(Q|E) \rightarrow P(Q|E,F) \rightarrow P(Q|E,F,G) \rightarrow \cdots$$

- Bayesian Inference: How beliefs *evolve* with observation
- Recursive: Posterior becomes prior of next inference step

General Bayesian Inference

- Observation of *additional* variables
  - Common case, e.g. image intensities, surrogate measures (size, …)
  - Coupled to core model via likelihood factorization
- General Bayesian inference case:
  - Distribution of data D (formerly Evidence)
  - Parameters θ (formerly Query)



$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} = \frac{P(D|\theta)P(\theta)}{\int P(D|\theta)P(\theta)d\theta}$$

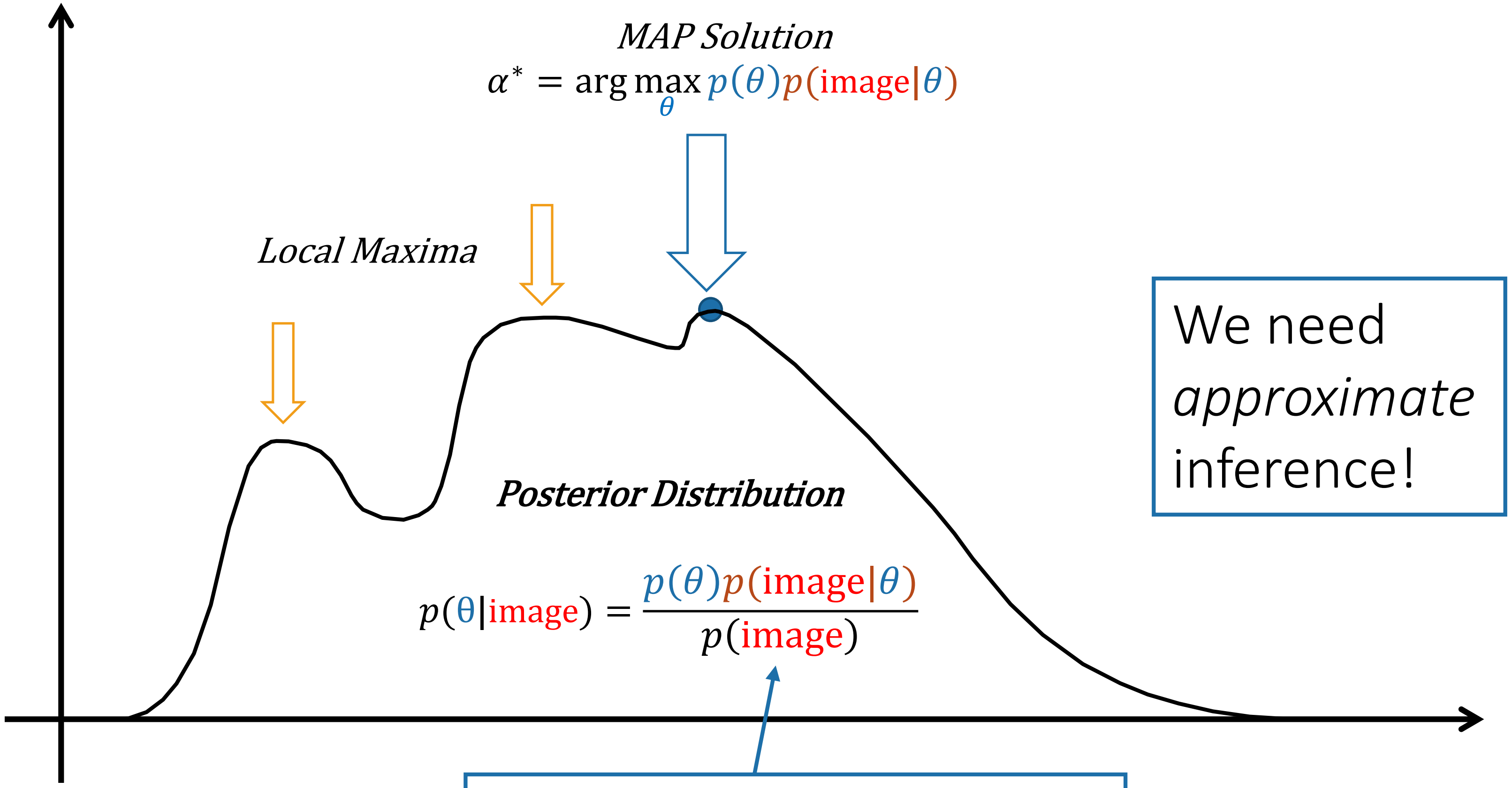
$$P(\theta|D) \propto P(D|\theta)P(\theta)$$

Checkpoint: Bayesian Inference

- Why is the Bayesian interpretation better suited for image analysis than a frequentist approach?
- Why is it often easier to specify a prior and a likelihood function, than the joint distribution?
- Bayesian inference can be applied recursively. Can you give an example (from the course) where we use the posterior again as a prior?
- Priors are subjective. Can we ever say one prior is better than another?
- Is it conceivable that two individuals assign mutually exclusive priors to the same situation
  - Can they ever converge to the same conclusion?

Fitting using Markov Chain Monte Carlo

Posterior distribution



Infeasible to compute:
$p(\text{image}) = \int p(\theta)p(\text{image}|\theta)\, d\theta$

We need *approximate* inference!

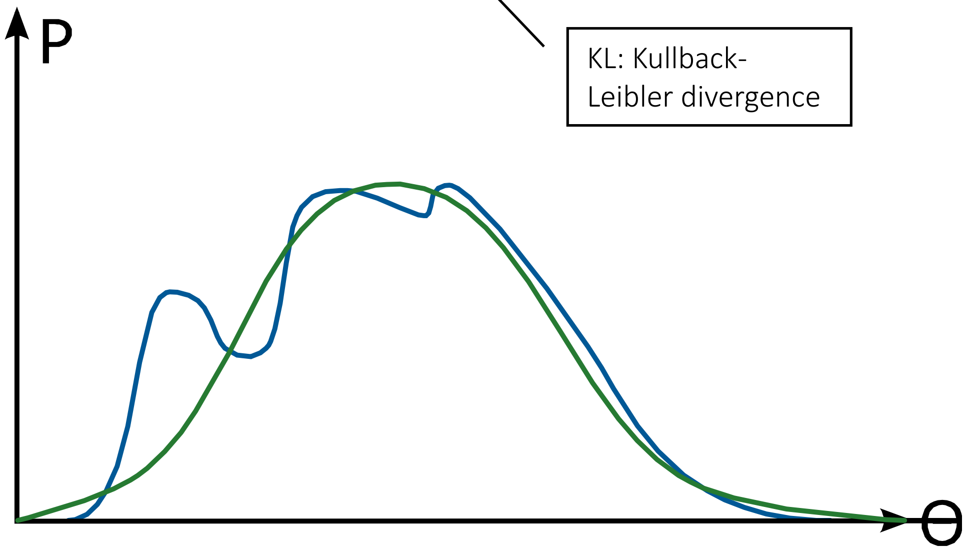
Approximate Bayesian Inference

Variational methods

- Function approximation $q(\theta)$

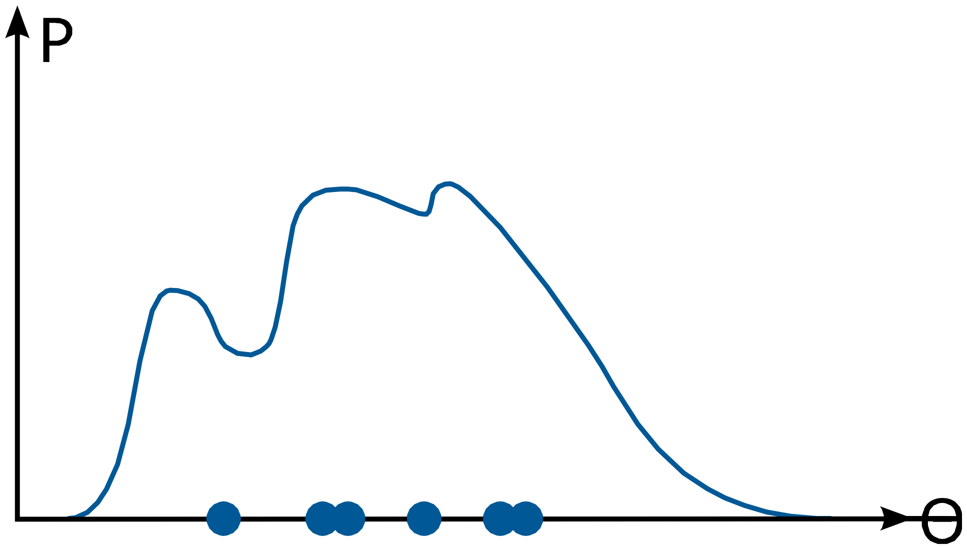
$$\arg\max_q \mathrm{KL}(q(\theta)|p(\theta|D))$$

KL: Kullback-Leibler divergence



Sampling methods

- Numeric approximations through simulation



Sampling Methods

- Simulate a distribution p through random samples x_i
- Evaluate expectations

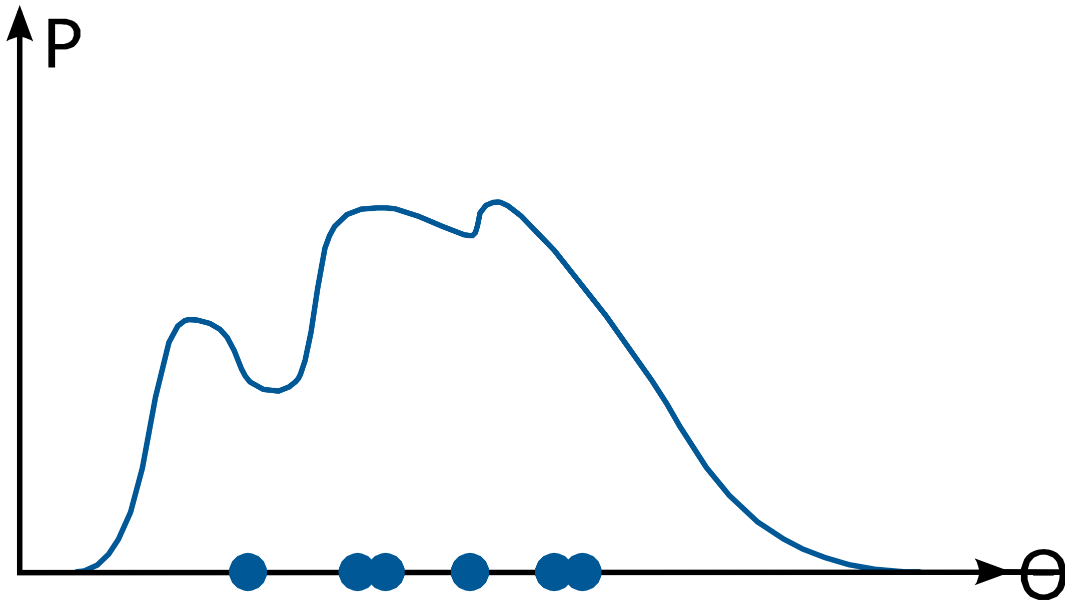
$$E[f(x)] = \int f(x)p(x)dx$$

$$E[f(x)] \approx \hat{f} = \frac{1}{N}\sum_{i}^{N} f(x_i), \qquad x_i \sim p(x)$$

This is difficult!

$$V[\hat{f}] \sim O\left(\frac{1}{N}\right)$$

- “Independent” of dimensionality
- More samples increase accuracy



Sampling from a Distribution

- Easy for standard distributions … is it?
  - Uniform
  - Gaussian

```
Random.nextDouble()
Random.nextGaussian()
```

- How to sample from more complex distributions?
  - Beta, Exponential, Chi square, Gamma, …
  - Posteriors are very often not in a “nice” standard text book form
- Sadly, only very few distributions are easy to sample from
  - We need to sample from an unknown posterior with only *unnormalized, expensive point-wise evaluation* ☹
- General Samplers?
  - Yes! – Rejection, Importance, *MCMC*

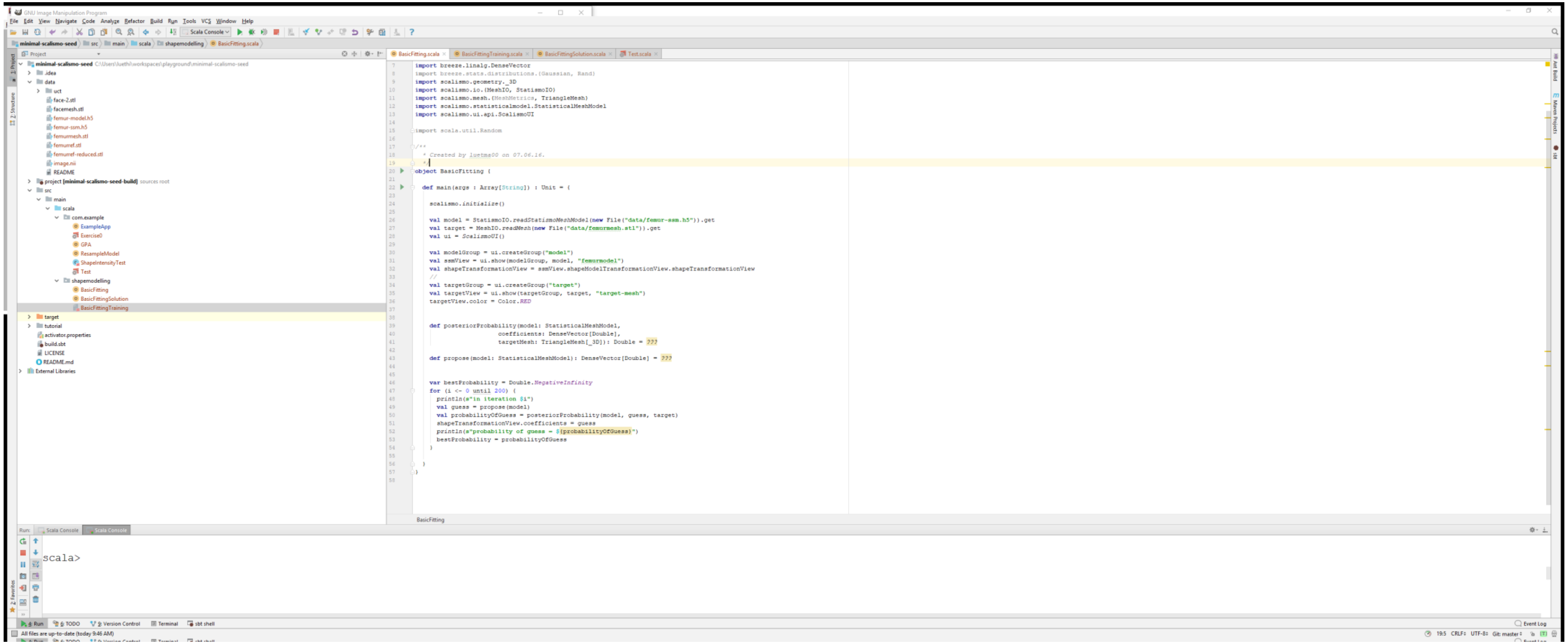
Markov Chain Monte Carlo

- Markov Chain Monte Carlo Methods (MCMC)

  Design a *Markov Chain* such that samples x obey the target distribution p

  Concept: *"Use an already existing sample to produce the next one"*

- Very powerful general sampling methods
  - Many successful practical applications
  - Proven: developed in the 1950/1970ies (Metropolis/Hastings)
  - Direct mapping of computing power to approximation accuracy
- Algorithms (buzz words):
  - Metropolis/-Hastings, Gibbs, Slice Sampling


BasicFitting.scala
BasicFittingTraining.scala
BasicFittingSolution.scala
Test.scala
import breeze.linalg.DenseVector
import breeze.stats.distributions.{Gaussian, Rand}
import scalismo.geometry._3D
import scalismo.io.{MeshIO, StatismoIO}
import scalismo.mesh.{MeshMetrics, TriangleMesh}
import scalismo.statisticalmodel.StatisticalMeshModel
import scalismo.ui.api.ScalismoUI
import scala.util.Random
/**
* Created by luetma00 on 07.06.16.
*/
object BasicFitting {
def main(args : Array[String]) : Unit = {
scalismo.initialize()
val model = StatismoIO.readStatismoMeshModel(new File("data/femur-ssm.h5")).get
val target = MeshIO.readMesh(new File("data/femurmesh.stl")).get
val ui = ScalismoUI()
val modelGroup = ui.createGroup("model")
val ssmView = ui.show(modelGroup, model, "femurmodel")
val shapeTransformationView = ssmView.shapeModelTransformationView.shapeTransformationView
//
val targetGroup = ui.createGroup("target")
val targetView = ui.show(targetGroup, target, "target-mesh")
targetView.color = Color.RED
def posteriorProbability(model: StatisticalMeshModel,
coefficients: DenseVector[Double],
targetMesh: TriangleMesh[_3D]): Double = ???
def propose(model: StatisticalMeshModel): DenseVector[Double] = ???
var bestProbability = Double.NegativeInfinity
for (i <- 0 until 200) {
println(s"in iteration $i")
val guess = propose(model)
val probabilityOfGuess = posteriorProbability(model, guess, target)
shapeTransformationView.coefficients = guess
println(s"probability of guess = ${probabilityOfGuess}")
bestProbability = probabilityOfGuess
}
}
}
BasicFitting
scala>

The Metropolis Algorithm

Requirements:

- Proposal distribution $Q(\boldsymbol{x}'|\boldsymbol{x})$ – *must generate samples, symmetric*
- Target distribution $P(\boldsymbol{x})$ – *with point-wise evaluation*

Result:

- Stream of samples approximately from $P(\boldsymbol{x})$

- Initialize with sample $\boldsymbol{x}$
- Generate next sample, with current sample $\boldsymbol{x}$
    1. Draw a sample $\boldsymbol{x}'$ from $Q(\boldsymbol{x}'|\boldsymbol{x})$ ("proposal")
    2. With *probability* $\alpha = \min\left\{\frac{P(\boldsymbol{x}')}{P(\boldsymbol{x})}, 1\right\}$ accept $\boldsymbol{x}'$ as new state $\boldsymbol{x}$
    3. Emit current state $\boldsymbol{x}$ as sample

Example: 2D Gaussian

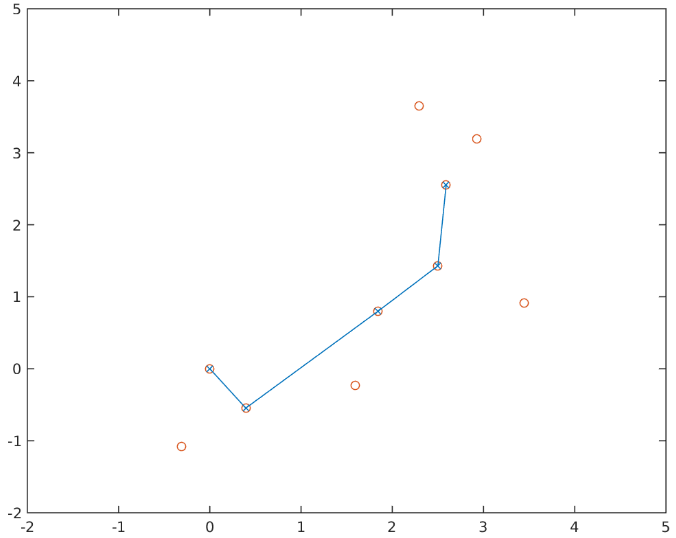
- Target: $P(\boldsymbol{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}$
- Proposal: $Q(\boldsymbol{x}'|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}'\,|\,\boldsymbol{x}, \sigma^2 I_2)$ ← Random walk

Target

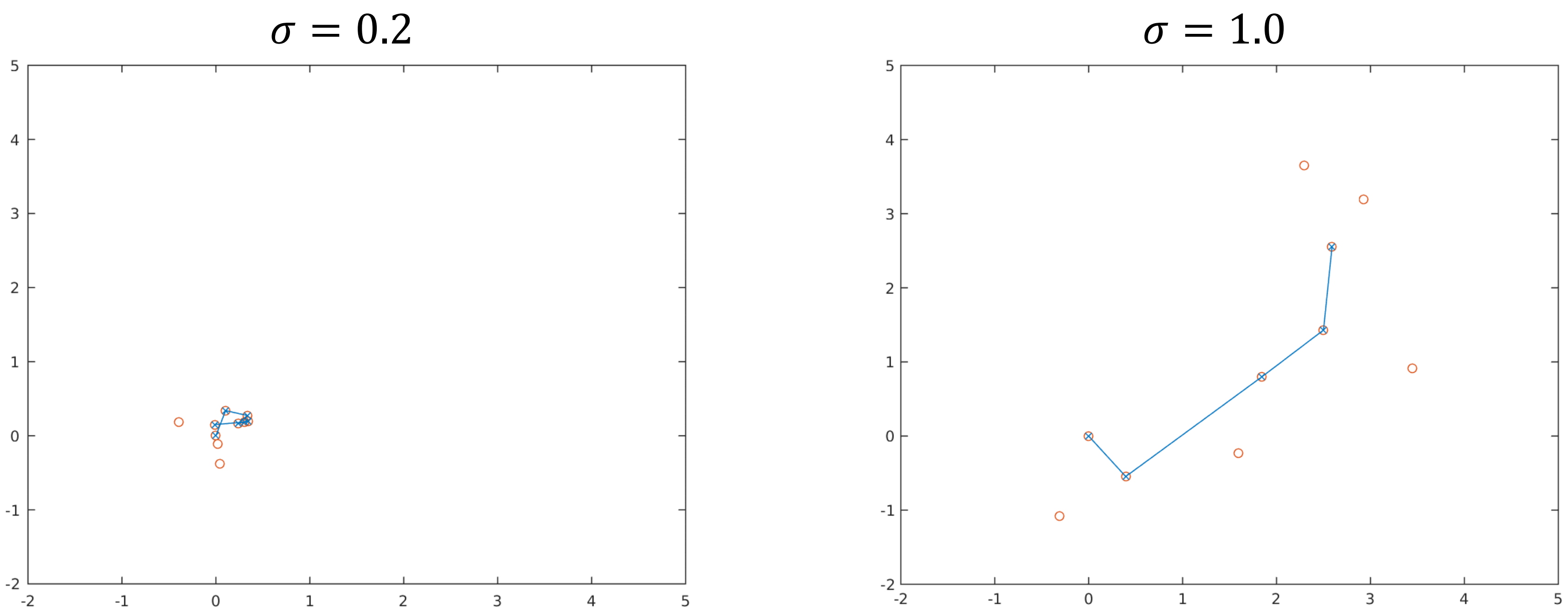
$$\mu = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$$
$$\Sigma = \begin{bmatrix} 1.25 & 0.75 \\ 0.75 & 1.25 \end{bmatrix}$$

Sampled Estimate

$$\hat{\mu} = \begin{bmatrix} 1.56 \\ 1.68 \end{bmatrix}$$
$$\hat{\Sigma} = \begin{bmatrix} 1.09 & 0.63 \\ 0.63 & 1.07 \end{bmatrix}$$



2D Gaussian: Different Proposals



The Metropolis-Hastings Algorithm

- Initialize with sample $\boldsymbol{x}$
- Generate next sample, with current sample $\boldsymbol{x}$
  1. Draw a sample $\boldsymbol{x}'$ from $Q(\boldsymbol{x}'|\boldsymbol{x})$ ("proposal")
  2. With *probability* $\alpha = \min\left\{\frac{P(x')}{P(x)}\frac{Q(x|x')}{Q(x'|x)}, 1\right\}$ accept $\boldsymbol{x}'$ as new state $\boldsymbol{x}$
  3. Emit current state $\boldsymbol{x}$ as sample

- Generalization of Metropolis algorithm to asymmetric Proposal distribution

$$Q(\boldsymbol{x}'|\boldsymbol{x}) \neq Q(\boldsymbol{x}|\boldsymbol{x}')$$

$$Q(\boldsymbol{x}'|\boldsymbol{x}) > 0 \Leftrightarrow Q(\boldsymbol{x}|\boldsymbol{x}') > 0$$

Properties

- **Approximation:** Samples $x_1, x_2, \ldots$ approximate $P(x)$

  Unbiased but correlated (not *i.i.d.*)

- **Normalization:** $P(x)$ does not need to be normalized

  Algorithm only considers ratios $P(x')/P(x)$

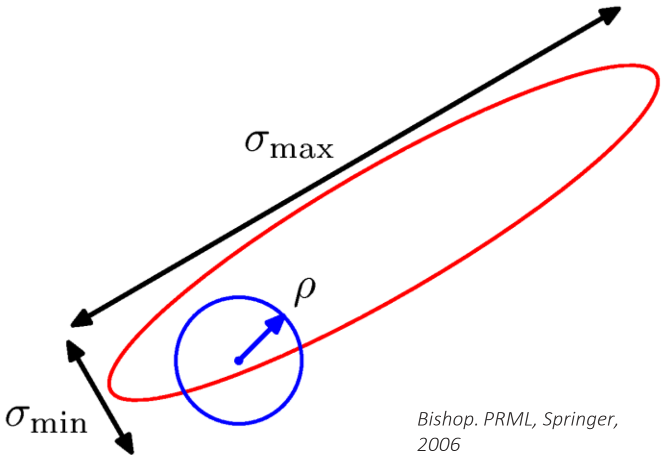
- **Dependent Proposals:** $Q(x'|x)$ depends on current sample x

  Algorithm adapts to target with simple 1-step memory

Metropolis - Hastings: Limitations

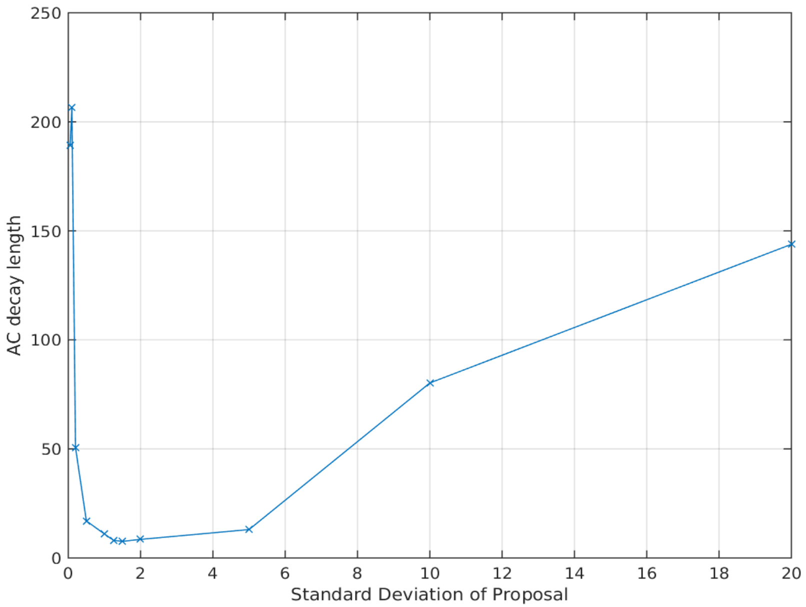
- Highly correlated targets

  Proposal should match target to avoid too many rejections



Bishop. PRML, Springer, 2006

- Serial correlation
  - Results from rejection and too small stepping
  - Subsampling



Propose-and-Verify Algorithm

- Metropolis algorithm formalizes: *propose-and-verify*
- *Steps are completely independent.*

Propose

Draw a sample x' from $Q(x'|x)$

Verify

With *probability* $\alpha = \min\left\{\frac{P(x')}{P(x)}\frac{Q(x|x')}{Q(x'|x)}, 1\right\}$ accept $\boldsymbol{x}'$ as new sample

MH as Propose and Verify

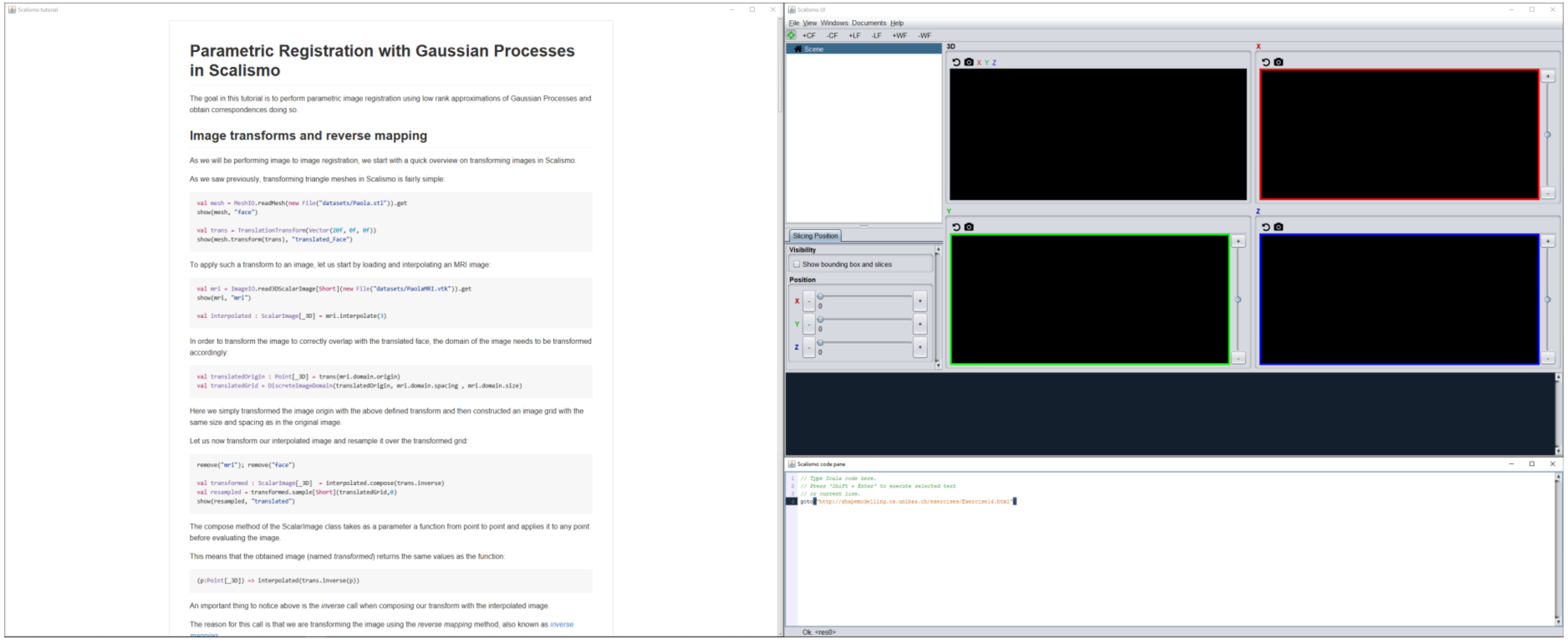
- Decouples the steps of finding the solution from validating a solution
- Natural to integrate uncertain proposals Q
  (e.g. automatically detected landmarks, ...)
- Possibility to include "local optimization" (e.g. a ASM or AAM updates, gradient step, ...) as proposal
  - Requires slight extension of the MH algorithm to avoid biased posterior.

Anything more "informed" than random walk should improve convergence.

Checkpoint: MCMC

- Why is it important in our model fitting problem, that the MH-algorithm can work with unnormalized distributions?
- Compare a classical (gradient-based) optimization algorithm to the MH-algorithm. How can the MH-Algorithm avoid getting stuck in local optima?
- What can you say about the samples coming from the MH-Algorithm
- Explain why choosing the proposals is very important for a good performance of the algorithm.

Exercise: MCMC in Scalismo



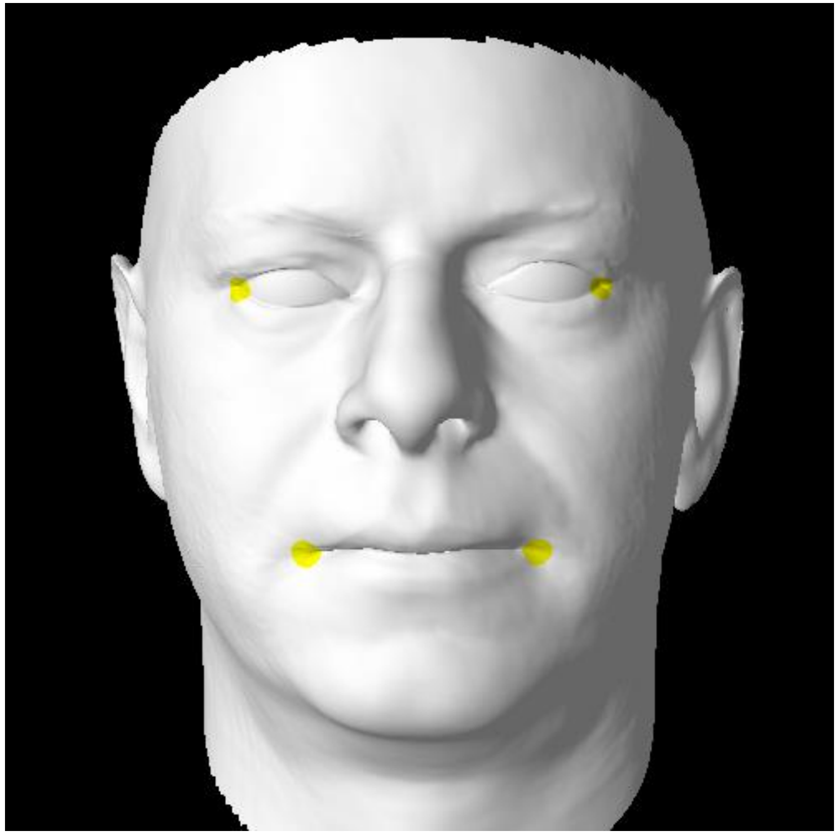
Type into the codepane:
`goto(“http://shapemodelling.cs.unibas.ch/exercises/Exercise15.html”)`

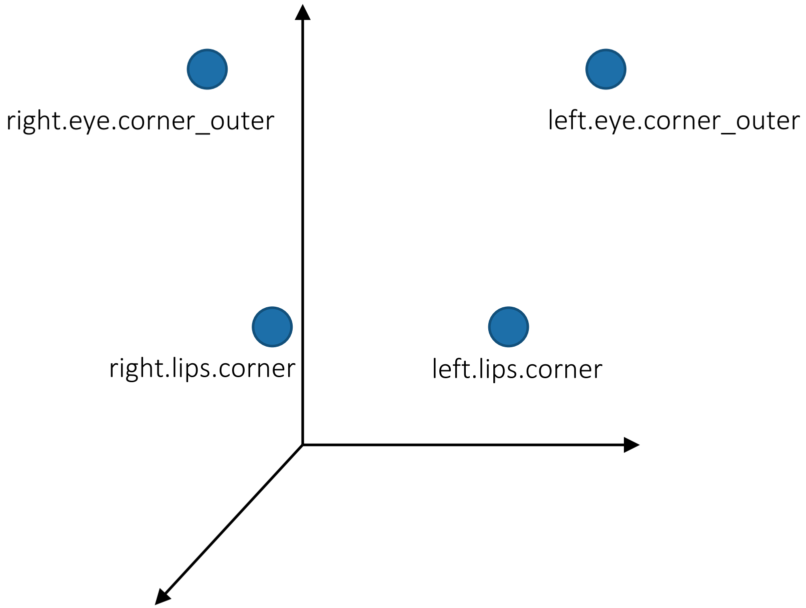
Scalismo 0.16: Check examples in `https://github.com/unibas-gravis/pmm2018`

Fitting 3D Landmarks

3D Alignment with Shape and Pose

3D Fitting Example





3D Fitting Setup

- 3D face model
- Arbitrary rigid transformation

  Pose, Positioning in space
- Observations
  - Observed positions $l_T^1, \dots, l_T^n$
  - Correspondence: $l_R^1, \dots, l_R^n$
- Goal: Find Posterior Distribution

$$P(\theta \middle| l_T^1, \dots, l_T^n) \propto p(l_T^1, \dots, l_T^R | \theta) P(\theta)$$

Parameters

$$\theta = (\alpha, \varphi, \psi, \vartheta, t)$$

Shape transformation

$$\varphi_S[\alpha] = \mu(x) + \sum_{i=1}^{r} \alpha_i \sqrt{\lambda_i} \Phi_i(x)$$

Rigid transformation

- 3 angles (pitch, yaw, roll) φ, ψ, ϑ
- Translation $t = (t_x, t_y, t_z)$

$$\varphi_R[\varphi, \psi, \vartheta, t] = R_\vartheta R_\psi R_\varphi(\boldsymbol{x}) + t$$

Full transformation

$$\varphi[\theta](x) = (\varphi_R \circ \varphi_S)[\theta](x)$$

Proposals

- Choose simple Gaussian random walk proposals (Metropolis)

$$"Q(\theta'|\theta) = N(\theta'|\theta, \Sigma_\theta)"$$

  - Normal *perturbations* of current state
- Block-wise to account for different parameter types
  - Shape $N(\boldsymbol{\alpha}'|\boldsymbol{\alpha}, \sigma_S^2 I_{m\times m})$
  - Rotation $N(\varphi'|\varphi, \sigma_\varphi^2)$, $N(\psi'|\psi, \sigma_\psi^2)$, $N(\vartheta'|\vartheta, \sigma_\vartheta^2)$
  - Translation $N(\boldsymbol{t}'|\boldsymbol{t}, \sigma_t^2 I_{3\times 3})$
- Large mixture distributions as proposals
  - Choose proposal Q_i with probability c_i

$$Q(\theta'|\theta) = \sum c_i Q_i(\theta'|\theta)$$

3DMM Landmarks Likelihood

Simple models: **Independent Gaussians**

Observation of L landmark locations l_T^i in image

- Single *landmark position* model:

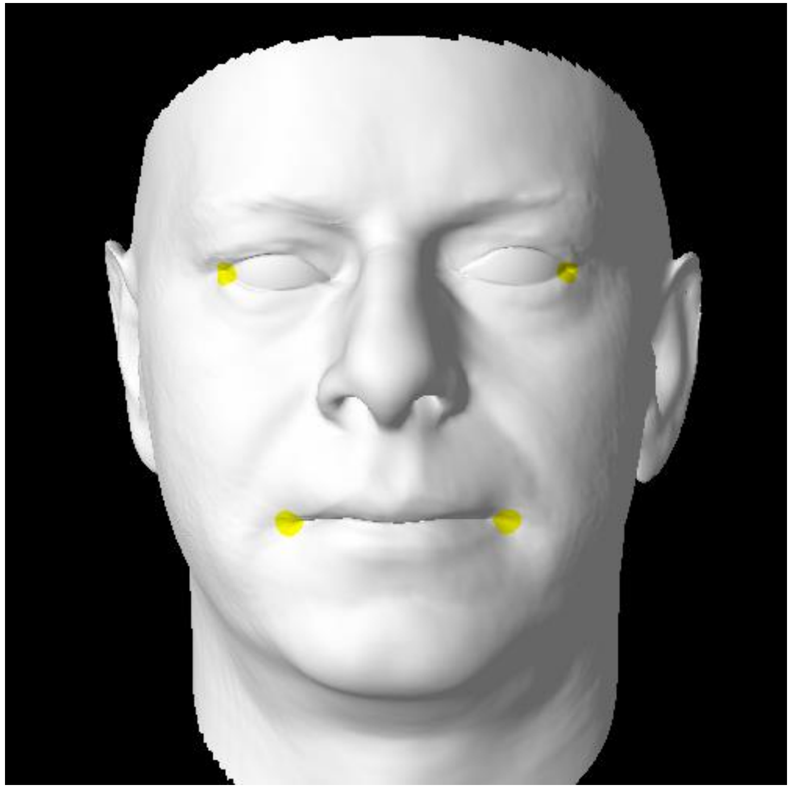
$$p(l_T|\theta, l_R) = N(\varphi[\theta](l_R), I_{3\times3}\sigma^2)$$

- *Independent* model (conditional independence):

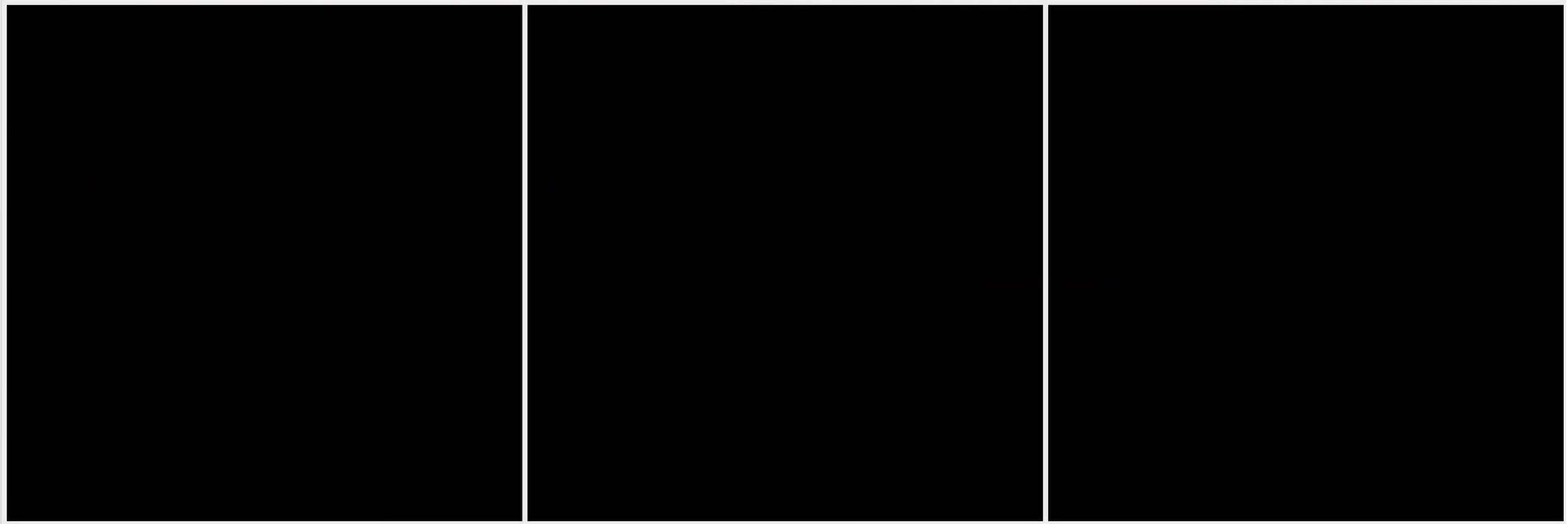
$$p(l_T^1, \ldots, l_T^n|\theta) = \prod_{i=1}^{L} p_i(l_T^i|\theta)$$

3D Fit to Landmarks

- Influence of landmarks uncertainty on final posterior?
  - $\sigma_{\mathrm{LM}} = 1\mathrm{mm}$
  - $\sigma_{\mathrm{LM}} = 4\mathrm{mm}$
  - $\sigma_{\mathrm{LM}} = 10\mathrm{mm}$
- Only 4 landmark observations:
  - Expect only weak shape impact
  - Should still constrain pose
- Uncertain LM should be looser



Posterior: Pose & Shape, 4mm



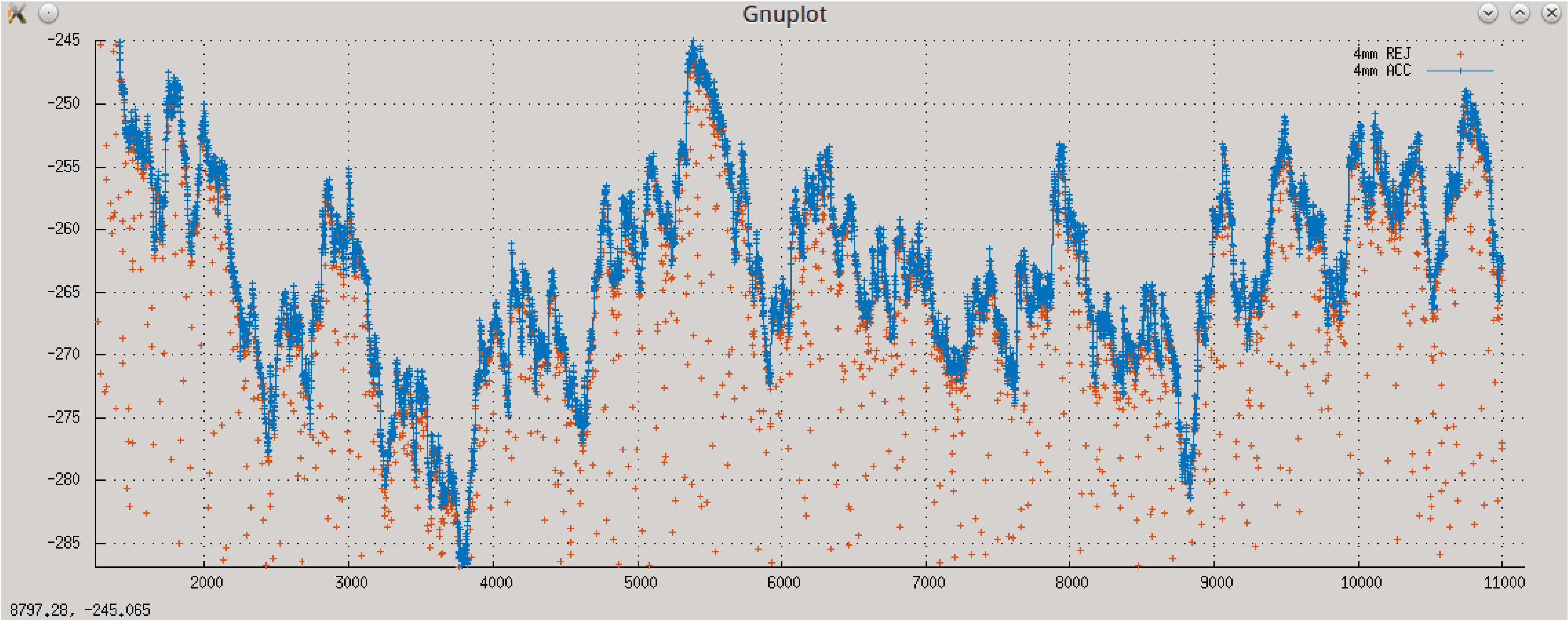
$\hat{\mu}_{\text{yaw}} = 0.511$ $\hat{\sigma}_{\text{yaw}} = 0.073\ (4°)$

$\hat{\mu}_{\text{t}_\text{x}} = -1\text{ mm}$ $\hat{\sigma}_{\text{t}_\text{x}} = 4\text{ mm}$

$\hat{\mu}_{\alpha_1} = 0.4$ $\hat{\sigma}_{\alpha_1} = 0.6$

(Estimation from samples)

Posterior: Pose & Shape, 4mm



Posterior values (log, unnormalized!)

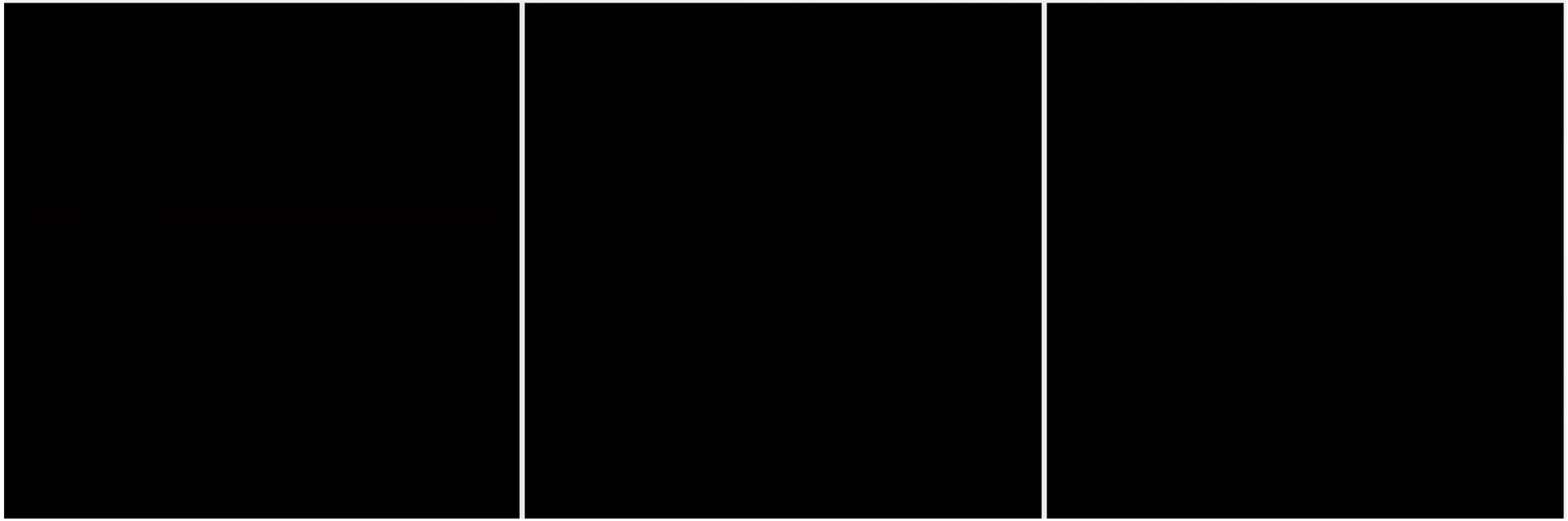
Posterior: Pose & Shape, 1mm

$\hat{\mu}_{\text{yaw}} = 0.50$ $\hat{\sigma}_{\text{yaw}} = 0.041\ (2.4°)$

$\hat{\mu}_{\text{t}_\text{x}} = -2\text{ mm}$ $\hat{\sigma}_{\text{t}_\text{x}} = 0.8\text{ mm}$

$\hat{\mu}_{\alpha_1} = 1.5$ $\hat{\sigma}_{\alpha_1} = 0.35$

Posterior: Pose & Shape, 10mm



$\hat{\mu}_{\text{yaw}} = 0.49$ $\quad$ $\hat{\mu}_{\text{t}_\text{x}} = -5\text{ mm}$ $\quad$ $\hat{\mu}_{\alpha_1} = 0$

$\hat{\sigma}_{\text{yaw}} = 0.11\ (7°)$ $\quad$ $\hat{\sigma}_{\text{t}_\text{x}} = 10\text{ mm}$ $\quad$ $\hat{\sigma}_{\alpha_1} = 0.6$

Summary: MCMC for 3D Fitting

- Probabilistic inference for fitting probabilistic models
  - Bayesian inference: posterior distribution
- Probabilistic inference is often intractable
  - Use *approximate* inference methods
- MCMC methods provide a powerful sampling framework
  - Metropolis-Hastings algorithm
    - Propose update step
    - Verify and accept with probability

- Samples converge to true distribution: More about this next time!