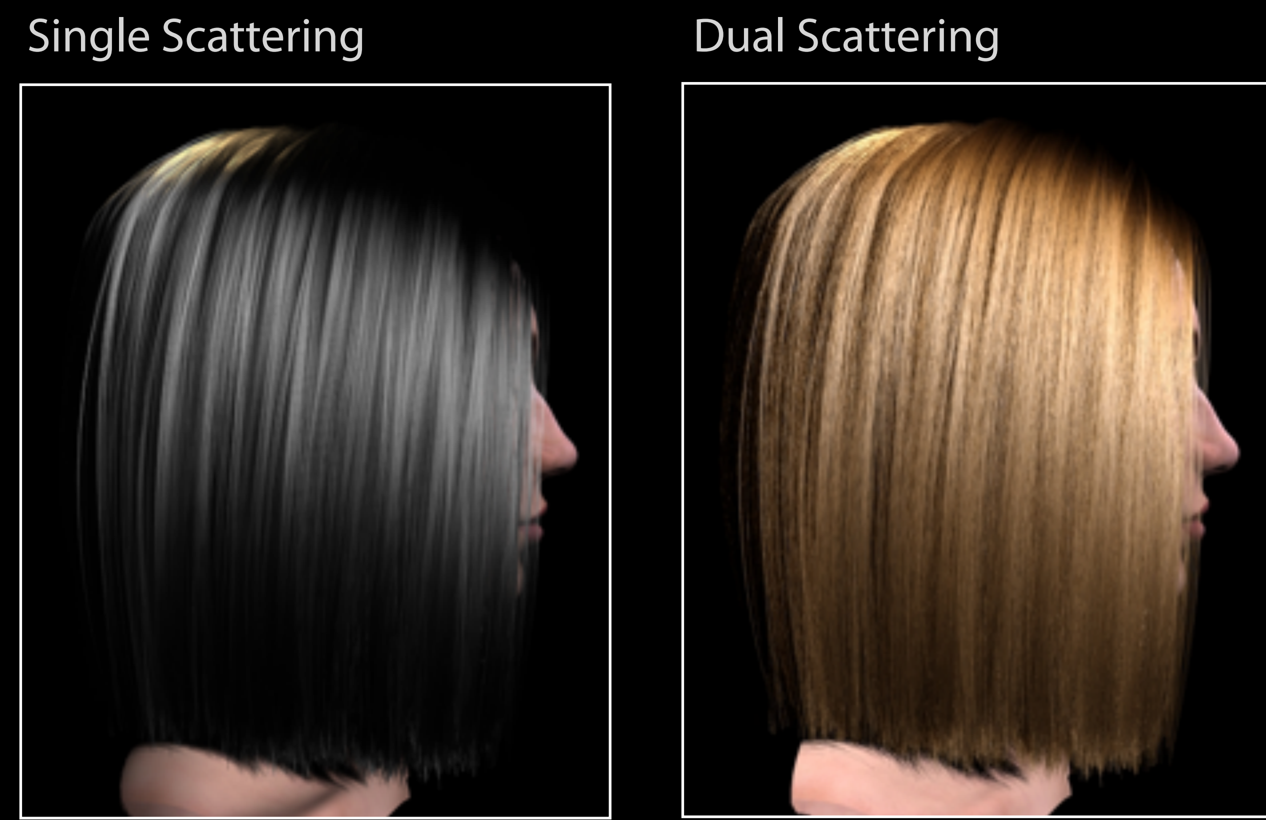
Cornell University

Capturing Hair Assemblies Fiber by Fiber

Wenzel Jakob
Jonathan T. Moon
Steve Marschner

Problem Statement

[Moon et al.]



[Zinke et al.]

Lack of realism in close up renderings

What is the physical arrangement of fibers in real hair?

Sunday, August 1, 2010

This project was started based on the observation that although today we have several efficient algorithms for rendering hair (while also accounting for multiple scattering), the resulting images often don't look that convincing -- an that is particularly true for for closeup renderings.

An important reason for this lack of realism is that the appearance of hair depends quite strongly on how the individual fibers are arranged locally.

In contrast to that, systems for generating hair geometry mostly focus on the large scale structure. They provide powerful tools for editing higher-level primitives, but the individual fibers are usually created using interpolation followed by the application of random perturbations, to make the hair look less regular.

The fiber arrangements generated with such tools look visually distinct from real hair and the consequence of this is that even with state-of-the art rendering methods, it is very challenging to produce renderings that remain convincing in close-up views.

Goal

- Capture accurate geometry of visible fibers within a small working volume

- Applications

  Ground truth verifications of rendering algorithms

  Better statistical distributions for hair modeling tools

  Detail transfer onto computer-generated models

In this talk, we present a system which acquires the geometry of fibers within a small working volume of about 6cm in diameter.

In contrast to previous systems, which have focused on capturing the large-scale structure of hairstyles on live subjects, this system aims to recover accurate geometry on much smaller scales, and is therefore most applicable to disembodied hair samples.
The captured data has several applications:
Apart from providing models for ground truth verifications, we also hope that it will lead to better statistical models used in hair modeling tools. Finally, an interesting application will be to transfer acquired detail-rich local structure onto coarser computer-generated models using texture synthesis techniques.

Outline

Prior work

Method concepts

Measurement setup

Data processing

Validation

Sunday, August 1, 2010

First, I'll go over some prior work and how it relates to this system.
Afterwards, I'll cover some of the fundamental ideas of the measurement approach and give an overview of the actual setup.
And finally, I'll show how the captured data is processed, followed by ground-truth comparisons.

Prior work — modeling & simulation

Modeling

• Kim and Neumann 2000, Sobottka et al. 2006	Surfaces
• Kim and Neumann 2002, Bertails et al. 2003	Multiresolution hierarchies
• Yu 2001, Hadap and Magnenat-Thalmann 2002	Advection through vector fields
• Wang et al. 2009	Detail transfer

Simulation

• Anjo et al. 1991, Choe and Ko 2005	Guide strands, Wisps
• Bertails et al. 2006	Rods
• Selle et al. 2008	Fibers

Research on hairstyle modeling has mostly focused on providing high-level primitives under user control. These include surfaces, multi-resolution hierarchies and vector fields. Common to almost all such approaches is the generation of fibers using random processes, which don't have much in common with the physical properties of real hair.

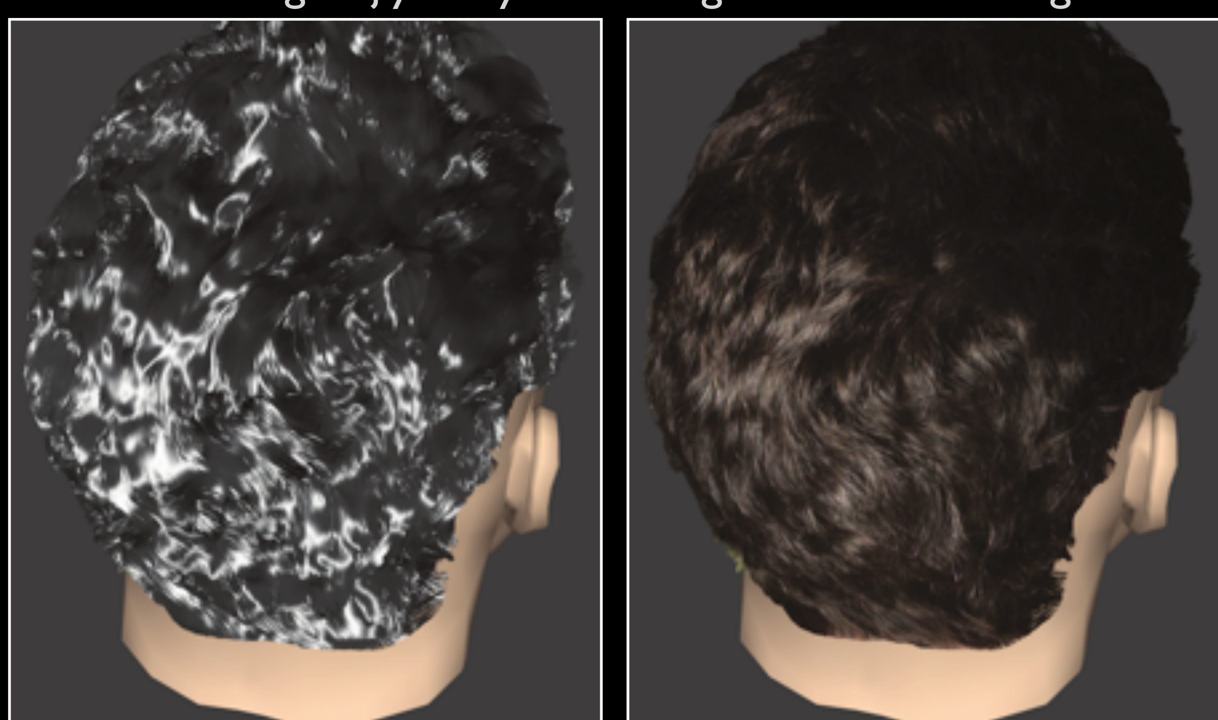
One interesting new direction is the application of texture synthesis techniques to combine parts of different hairstyles using detail transfer operations.

There have also been a number of works on simulating hairs. Recently, we have seen simulations which treat hair strands individually. The resulting models look very promising, but more work will be required to capture the full richness of physical interactions between hairs, to make them indistinguishable from real footage.

Prior work — acquisition

- Grabli et al. 2002 — Image-Based Hair Capture by Inverse Lighting
- Paris et al. 2004 — Capture of Hair Geometry from Multiple Images
- Wei et al. 2005 — Modeling Hair from Multiple Views
- Paris et al. 2008 — Hair Photobooth
- Bonneel et al. 2009 — Single Photo Estimation of Hair Appearance

Rendered using Kajiya-Kay   Image-based rendering



[Paris et al.]

Photography   Reconstruction

[Wei et al.]

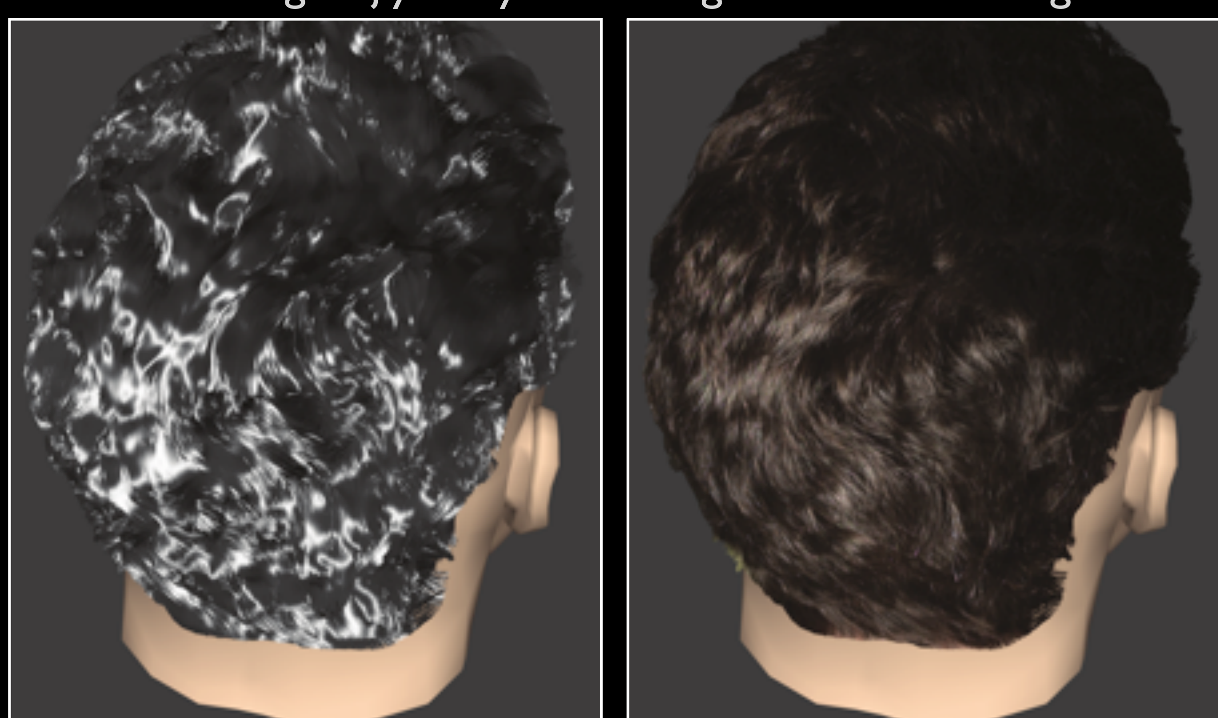
Sunday, August 1, 2010

On the acquisition side, the has been some research in estimating scattering model parameters. There are also several works, which acquire hairstyle geometry based on photographs. Common to all of them is that they don’t try to reconstruct the individual fibers. Instead, the input photographs are used to designate a volume to be filled with hair, and fibers are then added using chaining, interpolation or advection-based methods. I’d specifically like to highlight 2 papers here:

Prior work — acquisition

- Grabli et al. 2002 — Image-Based Hair Capture by Inverse Lighting
- Paris et al. 2004 — Capture of Hair Geometry from Multiple Images
- Wei et al. 2005 — Modeling Hair from Multiple Views
- **Paris et al. 2008 — Hair Photobooth**
- Bonneel et al. 2009 — Single Photo Estimation of Hair Appearance

Rendered using Kajiya-Kay | Image-based rendering



[Paris et al.]

Photography | Reconstruction

[Wei et al.]

Sunday, August 1, 2010

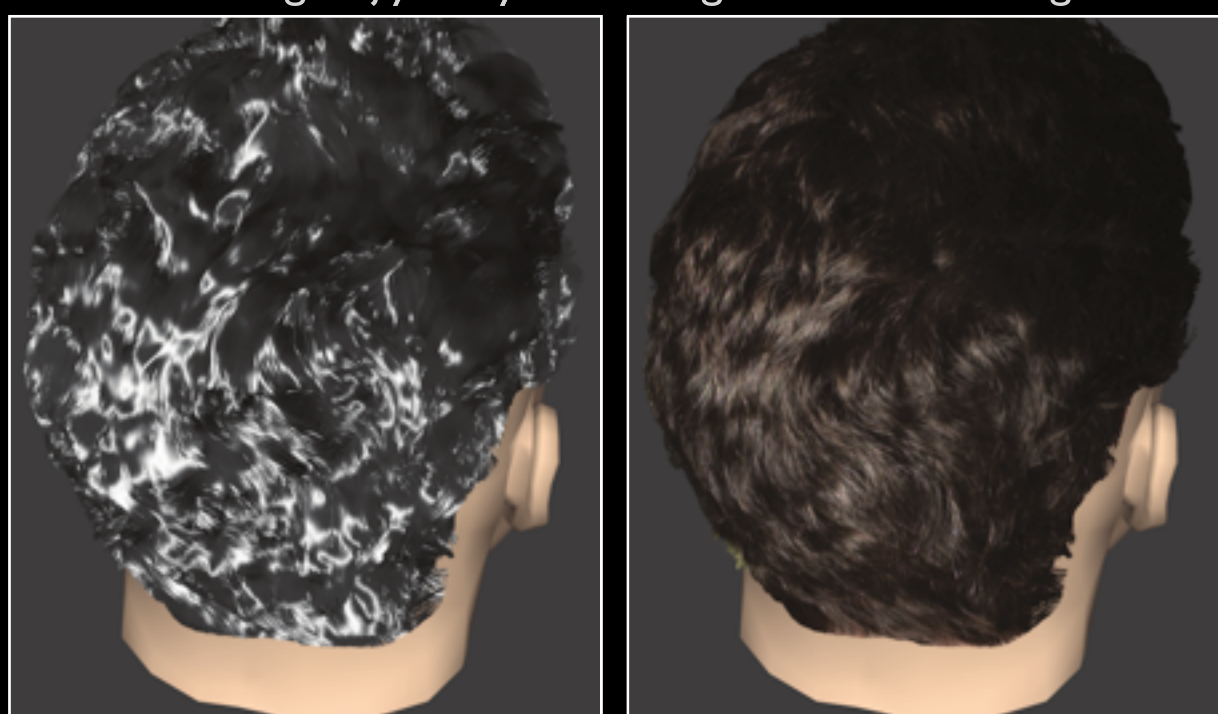
The 2008 “Hair Photobooth” paper presented a capture system based on a dome setup. The main problem we see with this approach is that, again, the reconstructed fibers are created using advection through a smooth vector field, and as a consequence, they mainly capture the large-scale structure. While the renderings created by this method are extremely convincing, it is important to not here that in order to achieve this high level of realism, image-based rendering methods are necessary.

In comparison, in this work, we want to generate models were the quality of the reconstruction makes this dependence unnecessary, so that the hair can be used in a physically-based render, without having to retain the input photographs.

Prior work — acquisition

- Grabli et al. 2002 — Image-Based Hair Capture by Inverse Lighting
- Paris et al. 2004 — Capture of Hair Geometry from Multiple Images
- Wei et al. 2005 — Modeling Hair from Multiple Views
- Paris et al. 2008 — Hair Photobooth
- Bonneel et al. 2009 — Single Photo Estimation of Hair Appearance

Rendered using Kajiya-Kay | Image-based rendering



[Paris et al.]

Photography | Reconstruction

[Wei et al.]

Sunday, August 1, 2010

The paper that is perhaps the most closely related to this work is by [Wei et al.] (click) from 2005. Here, fibers were traced through orientation fields captured using a handheld camera. But again, this method is not able to recreate the individual strands.

Method concepts

Sunday, August 1, 2010

Here is a picture of what our measurement setup looks like. It consists of a digital SLR that images a hair sample through a macro lens.
A large area light source is placed behind the camera, to provide illumination that is as uniform as possible. Additional precautions are to minimize vibration and air movement to a minimum, so that the hairs remain still during the course of a measurement.

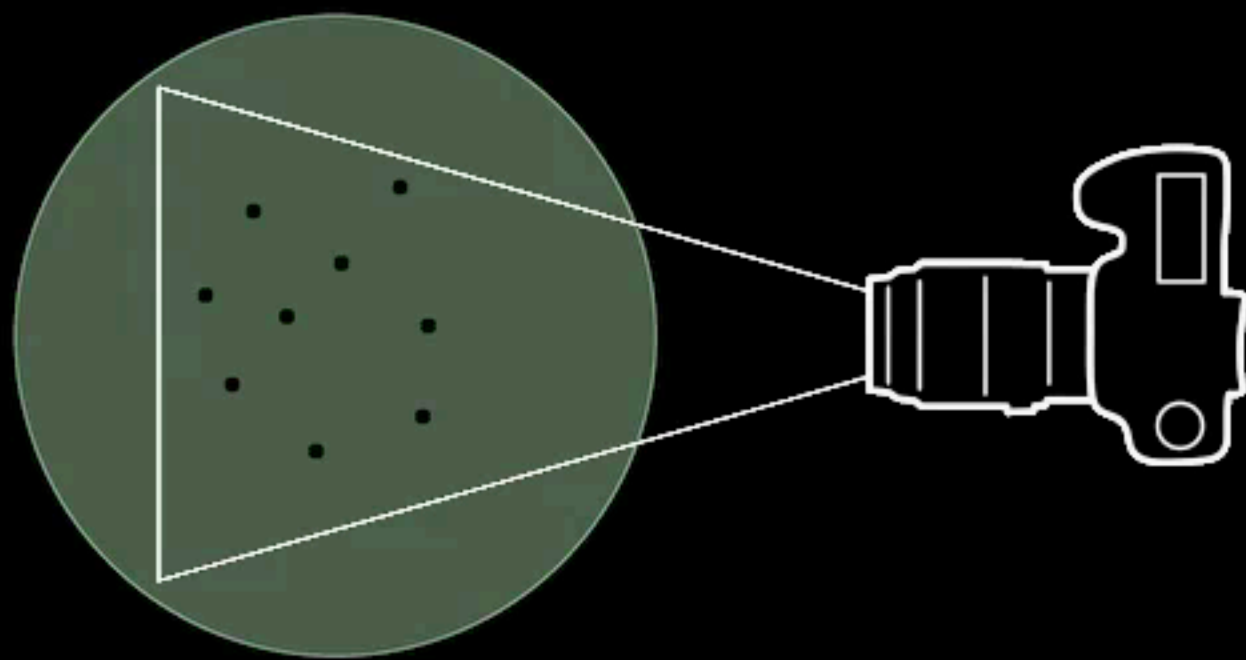
Method concepts

- Take many macrophotographs using a digital SLR mounted on a translation stage.
- Shallow depth of field: fibers come into focus as the focus plane passes them.

Before we continue, let’s first look at a high-level sketch, to understand the properties of this setup.
The camera is mounted on a translation stage, which lets us sweep the plane of focus while at the same time taking photographs at a densely spaced set of camera positions. (play)

Because we use a very shallow depth of field, most hairs in any image will be completely blurred out. Hairs that are very close to the plane of focus appear as bright lines, which makes their detection possible.
[play animation]
You will all agree that if we could pinpoint the precise place where the focus plane passes a fiber, then finding its 3D position is trivial – we simply record the observation’s pixel coordinates and translation depth, and that gives the 3D position.

Method concepts



- Take many macrophotographs using a digital SLR mounted on a translation stage.
- Shallow depth of field: fibers come into focus as the focus plane passes them.

Before we continue, let's first look at a high-level sketch, to understand the properties of this setup.
The camera is mounted on a translation stage, which lets us sweep the plane of focus while at the same time taking photographs at a densely spaced set of camera positions. (play)

Because we use a very shallow depth of field, most hairs in any image will be completely blurred out. Hairs that are very close to the plane of focus appear as bright lines, which makes their detection possible.
[play animation]
You will all agree that if we could pinpoint the precise place where the focus plane passes a fiber, then finding its 3D position is trivial – we simply record the observation's pixel coordinates and translation depth, and that gives the 3D position.

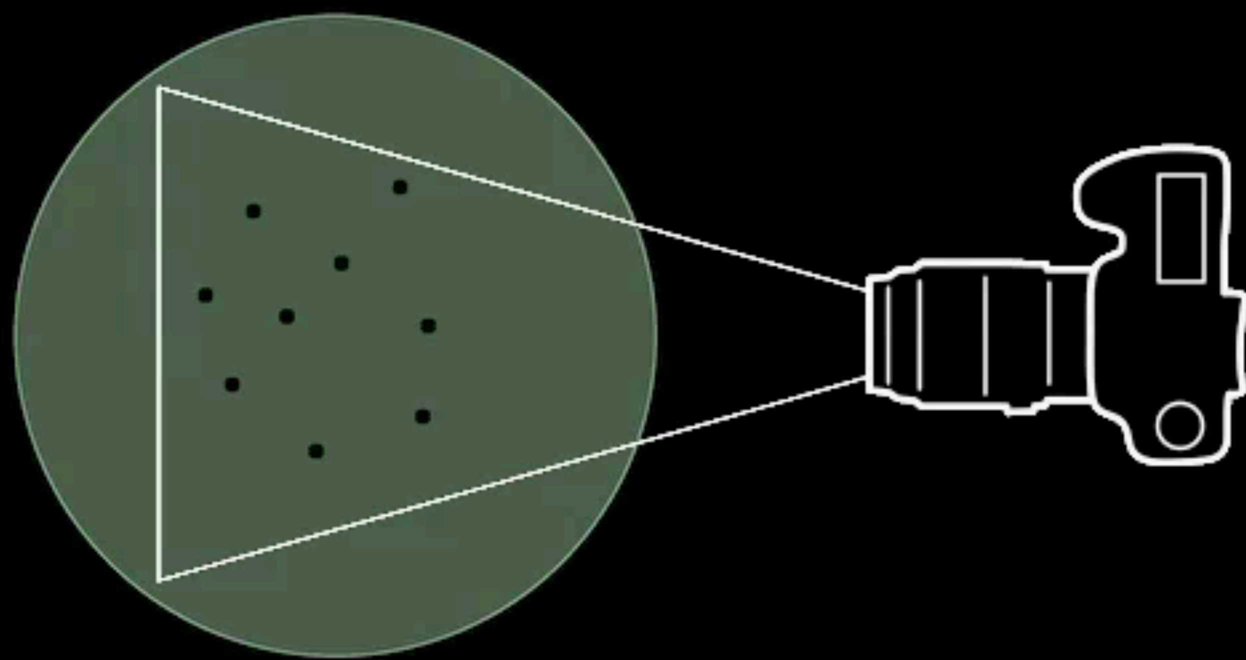
Method concepts

- Take many macrophotographs using a digital SLR mounted on a translation stage.
- Shallow depth of field: fibers come into focus as the focus plane passes them.

Before we continue, let's first look at a high-level sketch, to understand the properties of this setup.
The camera is mounted on a translation stage, which lets us sweep the plane of focus while at the same time taking photographs at a densely spaced set of camera positions.

Because we use a very shallow depth of field, most hairs in any image will be completely blurred out. Hairs that are very close to the plane of focus appear as bright lines, which makes their detection possible.
[play animation]
You will all agree that if we could pinpoint the precise place where the focus plane passes a fiber, then finding its 3D position is trivial – we simply record the observation's pixel coordinates and translation depth, and that gives the 3D position.

Method concepts



- Take many macrophotographs using a digital SLR mounted on a translation stage.
- Shallow depth of field: fibers come into focus as the focus plane passes them.

Before we continue, let's first look at a high-level sketch, to understand the properties of this setup.
The camera is mounted on a translation stage, which lets us sweep the plane of focus while at the same time taking photographs at a densely spaced set of camera positions.

Because we use a very shallow depth of field, most hairs in any image will be completely blurred out. Hairs that are very close to the plane of focus appear as bright lines, which makes their detection possible.
[play animation]
You will all agree that if we could pinpoint the precise place where the focus plane passes a fiber, then finding its 3D position is trivial – we simply record the observation's pixel coordinates and translation depth, and that gives the 3D position.

Method concepts

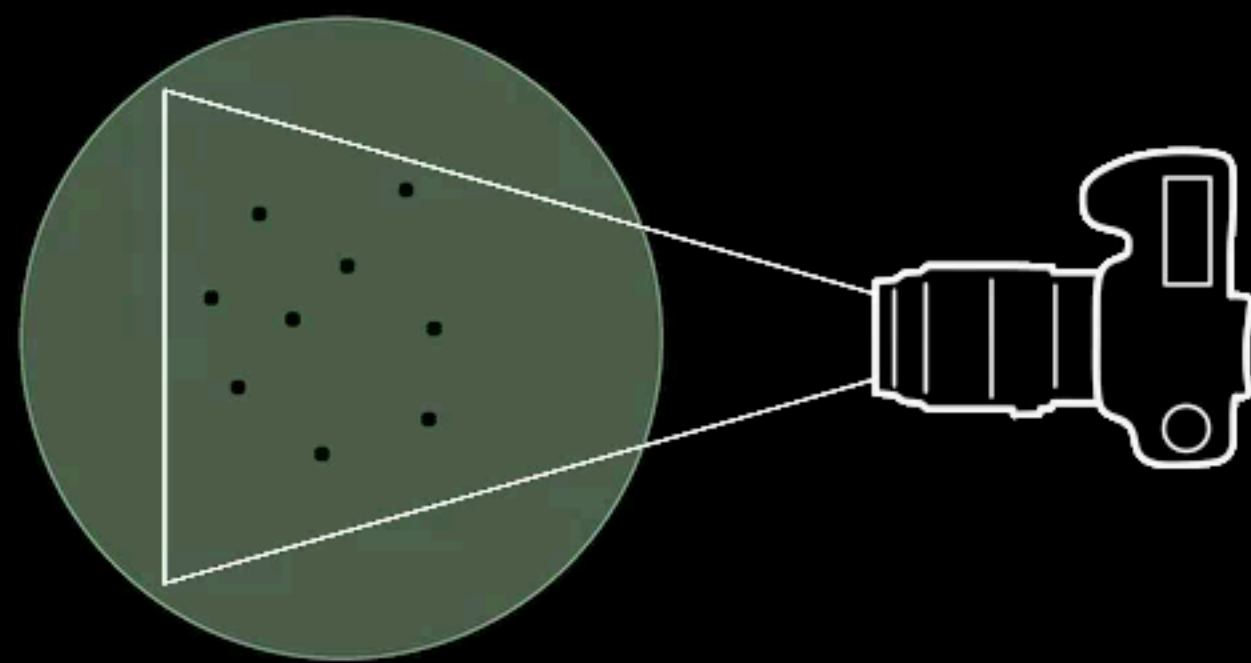
Issues with this approach:

- Uncertainty in depth due to filter response
- Problems with occlusion

There are several problems with this approach:
- the most obvious one, of course, is that we can only reconstruct hairs seen from a particular angle.
- also, hairs will often occlude each other from certain angles, and it would be beneficial to use information from other viewpoints to alleviate this.
- Most problematic, though, is the uncertainty in depth. (Play) Due to noise, non-diffuse reflection behavior of the hair and mutual occlusion, it is in general very hard to identify the exact point where
hair is most in focus.
- If we apply a filter that responds to well-focused hairs, it will actually do so for a whole region around the focus plane. This manifests itself as an uncertainty ellipse around the hair.

Method concepts

Issues with this approach:



- Uncertainty in depth due to filter response
- Problems with occlusion

Sunday, August 1, 2010

There are several problems with this approach:
 - the most obvious one, of course, is that we can only reconstruct hairs seen from a particular angle.
 - also, hairs will often occlude each other from certain angles, and it would be beneficial to use information from other viewpoints to alleviate this.
 - Most problematic, though, is the uncertainty in depth. (Play) Due to noise, non-diffuse reflection behavior of the hair and mutual occlusion, it is in general very hard to identify the exact point where
hair is most in focus.
 - If we apply a filter that responds to well-focused hairs, it will actually do so for a whole region around the focus plane. This manifests itself as an uncertainty ellipse around the hair.

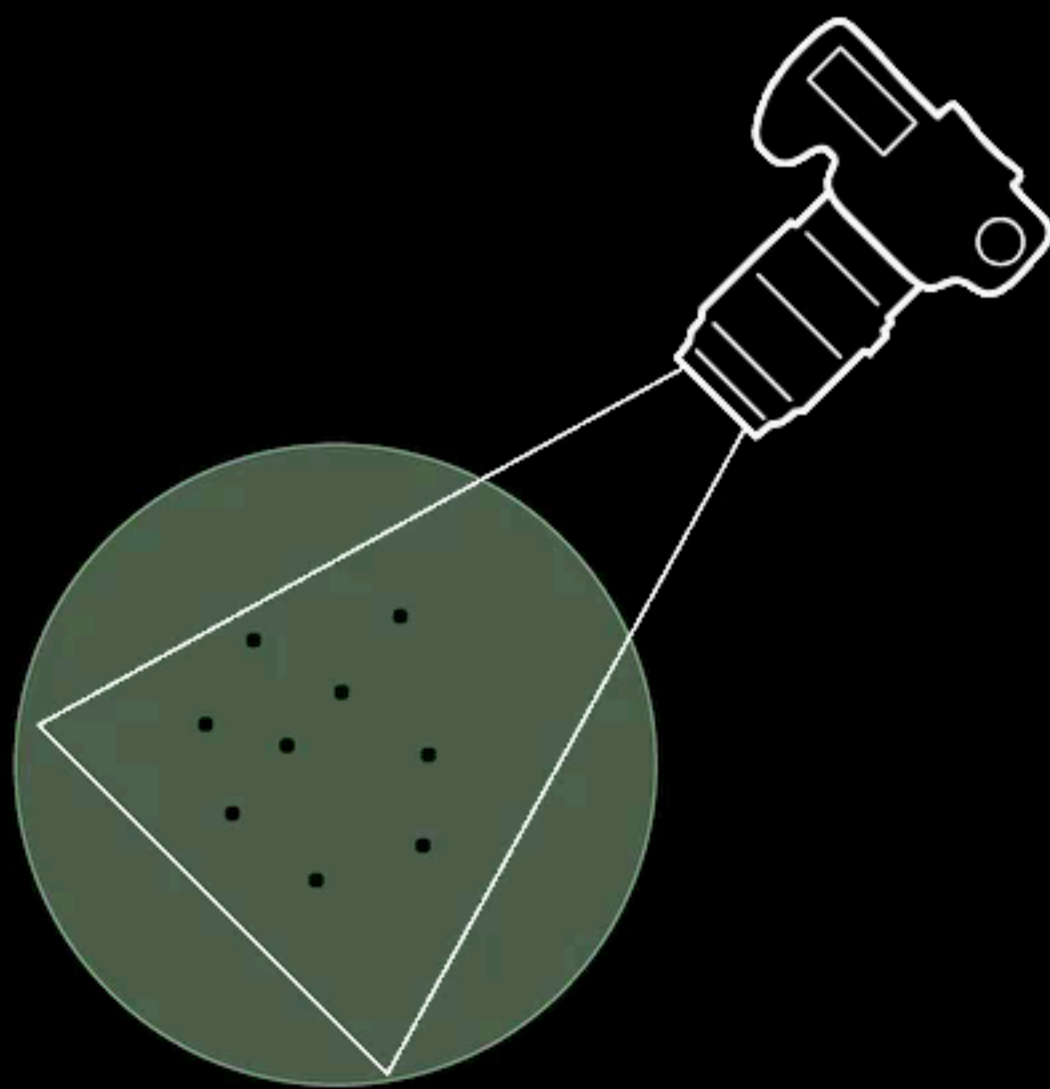
Method concepts

- Make use of complementary uncertainties in different views.

Sunday, August 1, 2010

One property of our measurement setup is that these uncertainty ellipses provide very accurate localization along the x axis of the camera and very poor localization along the translation axis. [Play] We make use of this by capturing sweeps from a number of different angles. The uncertainties are complementary and precisely pin down each hair. If we missed a hair from a particular angle, there is also a good chance that we will see it from somewhere else.

Method concepts



- Make use of complementary uncertainties in different views.

Sunday, August 1, 2010

One property of our measurement setup is that these uncertainty ellipses provide very accurate localization along the x axis of the camera and very poor localization along the translation axis. [Play] We make use of this by capturing sweeps from a number of different angles. The uncertainties are complementary and precisely pin down each hair. If we missed a hair from a particular angle, there is also a good chance that we will see it from somewhere else.

Setup

376 pictures / rotation
$\times$ 24 rotations
$=$ 9024 pictures

Timelapse: ~350x

Sunday, August 1, 2010

In practice, it is much easier to mount the hair on a rotation stage, instead of rotating the camera. Here you can see a time-lapse of a measurement in progress. [play] For a capture, we use 24 rotation angles and capture ~400 pictures per angular sweep, resulting in a total of 9024 pictures during a measurement. All of this takes between 6 and 12 hrs.

Setup

376 pictures / rotation
$\times$ 24 rotations
$=$ 9024 pictures

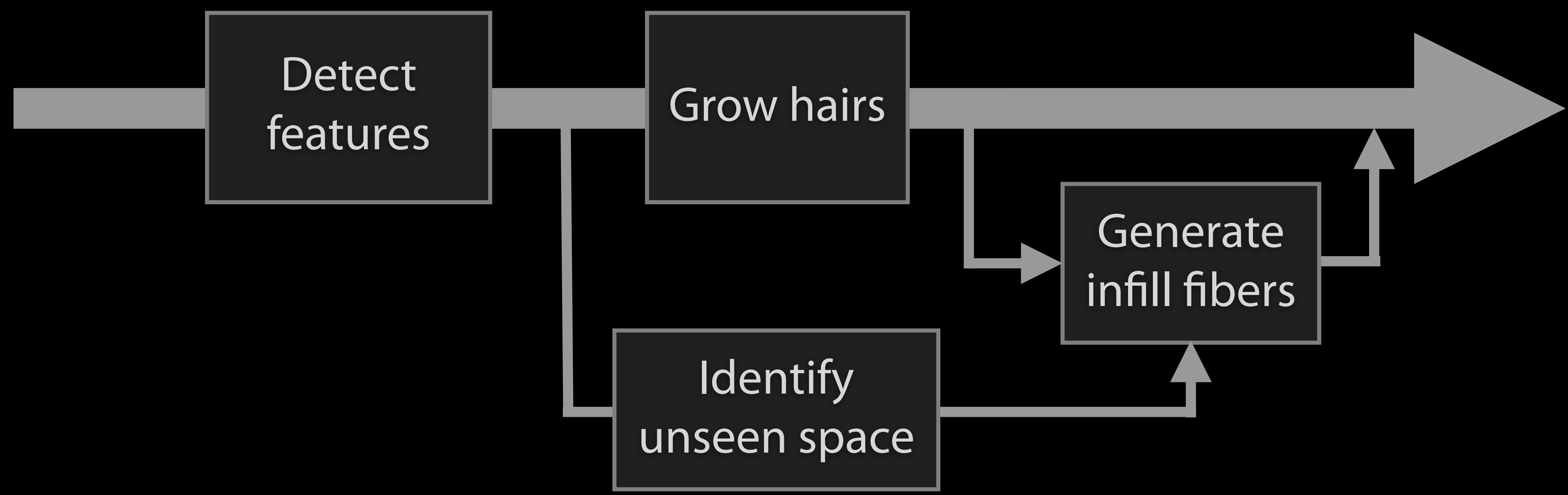
Sunday, August 1, 2010

In practice, it is much easier to mount the hair on a rotation stage, instead of rotating the camera. Here you can see a time-lapse of a measurement in progress. [play] For a capture, we use 24 rotation angles and capture ~400 pictures per angular sweep, resulting in a total of 9024 pictures during a measurement. All of this takes between 6 and 12 hrs.

Sunday, August 1, 2010

This is an animation of the photographs taken during one such sweep and you can see how each fiber comes into focus for a short amount of time. Next, I’ll show an overview of the processing pipeline we use.

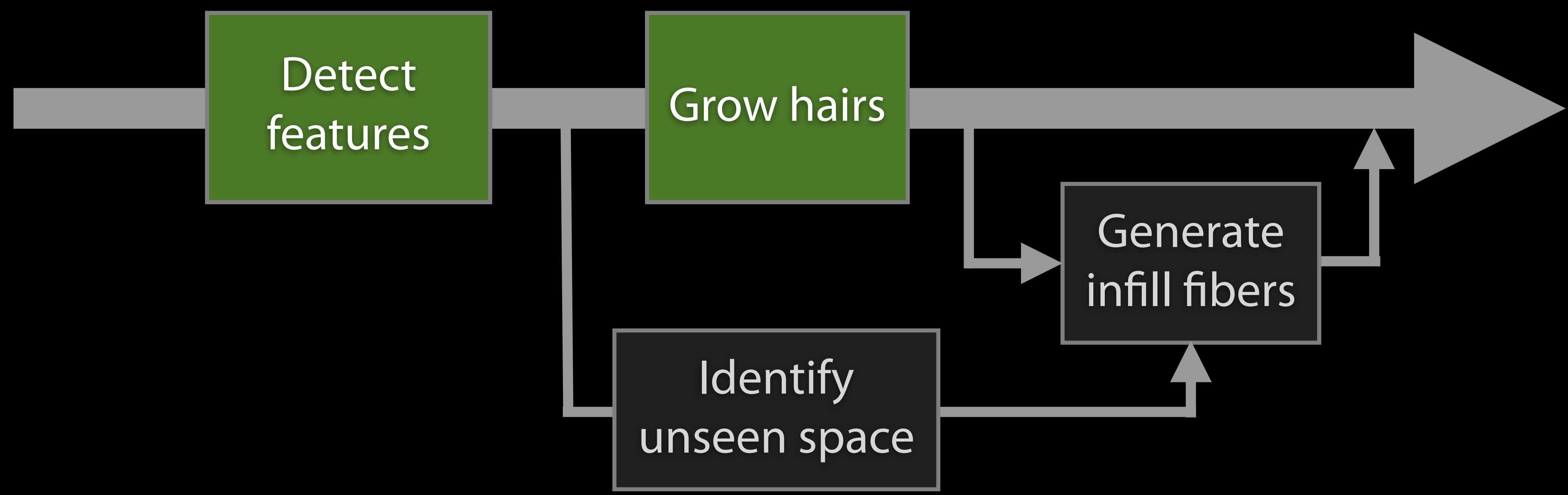
Pipeline overview



Sunday, August 1, 2010

I'll mainly talk about two parts, [click] namely the feature detection and fiber-growing stages. These are also the main contributions of the paper. I'll just quickly mention the parts that I won't have time to cover in detail: ....

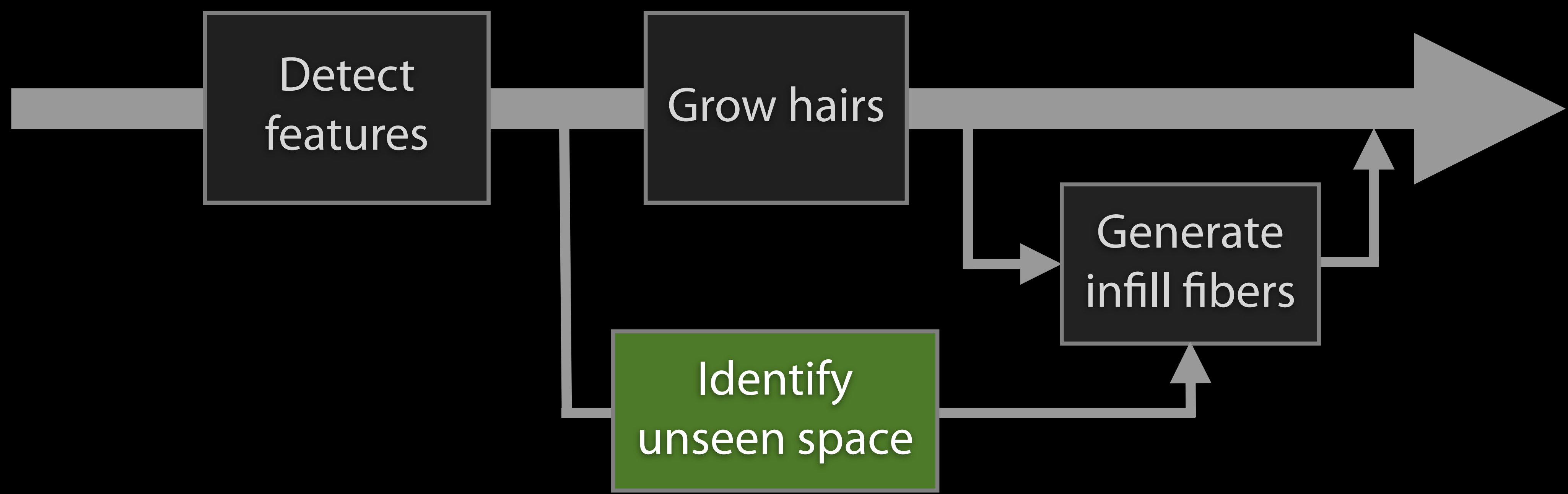
Pipeline overview



Sunday, August 1, 2010

I'll mainly talk about two parts, [click] namely the feature detection and fiber-growing stages. These are also the main contributions of the paper. I'll just quickly mention the parts that I won't have time to cover in detail: ....

Pipeline overview

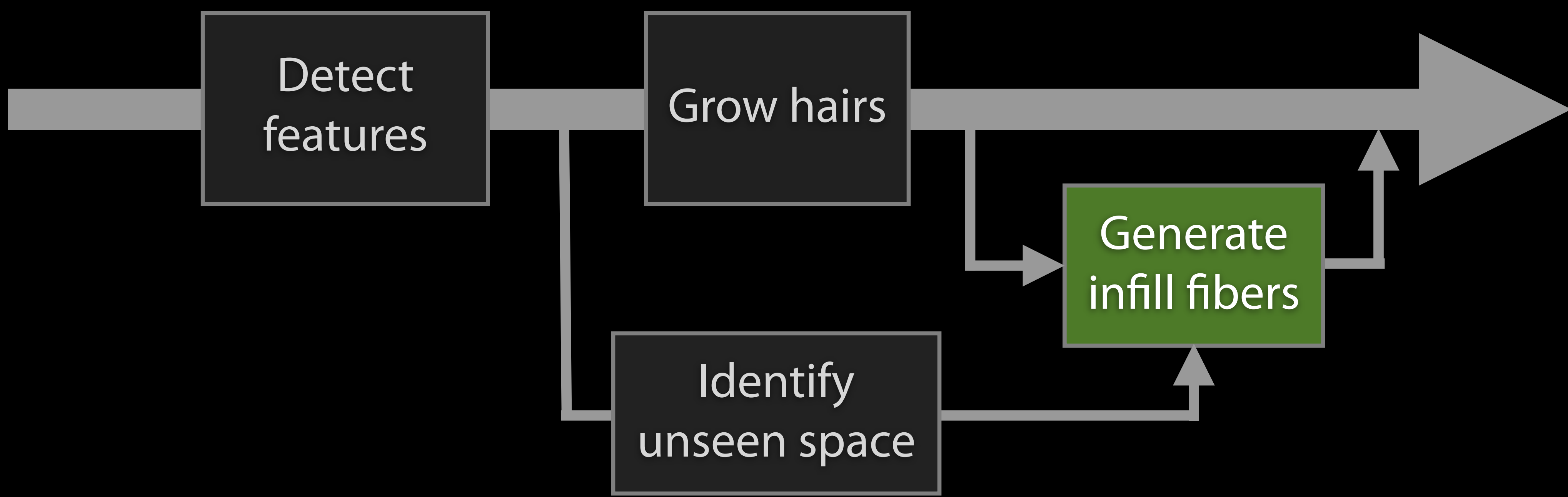


- Significant amounts of occlusion may prevent the reconstruction of fibers
- We propose a separate pipeline to detect affected regions

Sunday, August 1, 2010

Since our capture setup is based on photographs, there are regions where the occlusion is simply too strong to be able to reliably detect hair. For cases where this is undesirable, the paper contains additional information on detecting such regions and ...

Pipeline overview

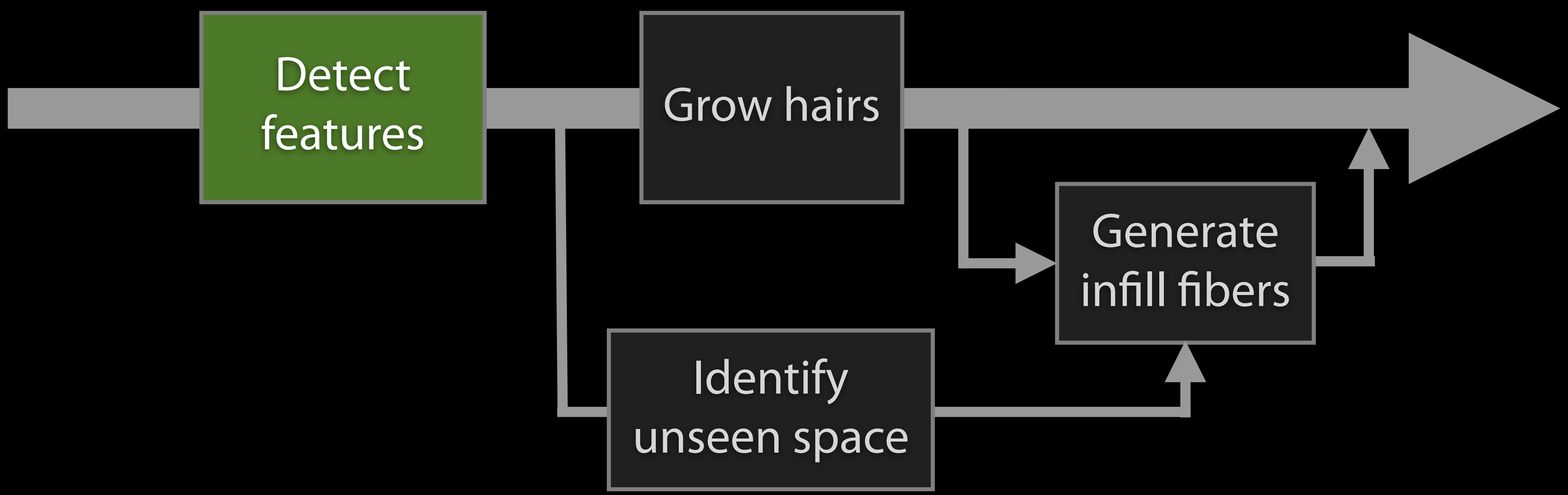


- Significant amounts of occlusion may prevent the reconstruction of fibers
- We propose a separate pipeline to detect affected regions
- Advection-based approach used as a fallback

Sunday, August 1, 2010

... on generating approximate infill fibers as a fallback process.

Pipeline overview

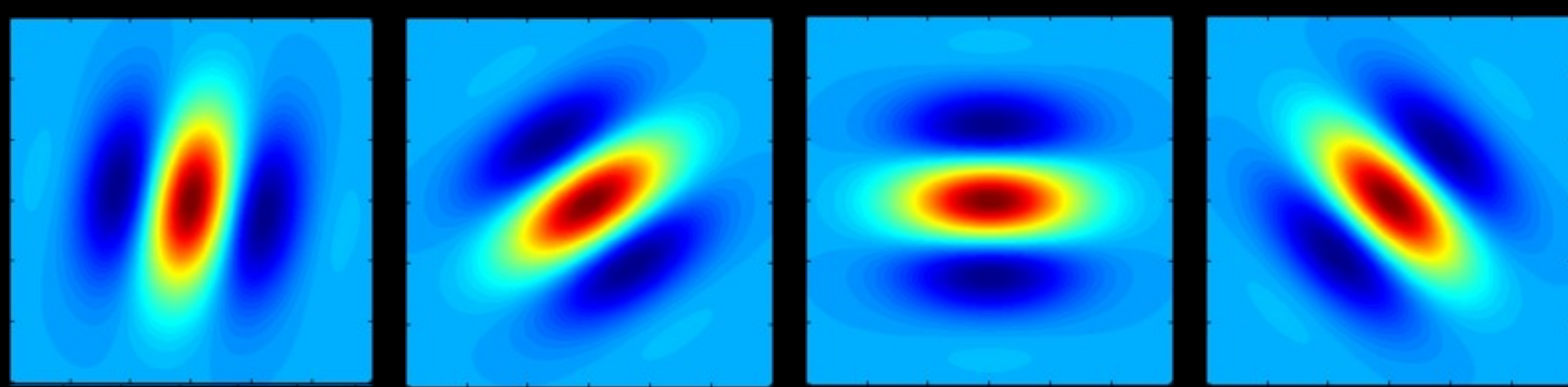


Sunday, August 1, 2010

Let’s start with feature detection.

Feature detection

- Finding hairs $\widehat{=}$ detecting ridges
- Modified Canny filter (scale space formulation)
- Filter bank of 32 7x7 Gabor filter kernels
  (these provide higher-accuracy 2D orientation information for use in later stages)



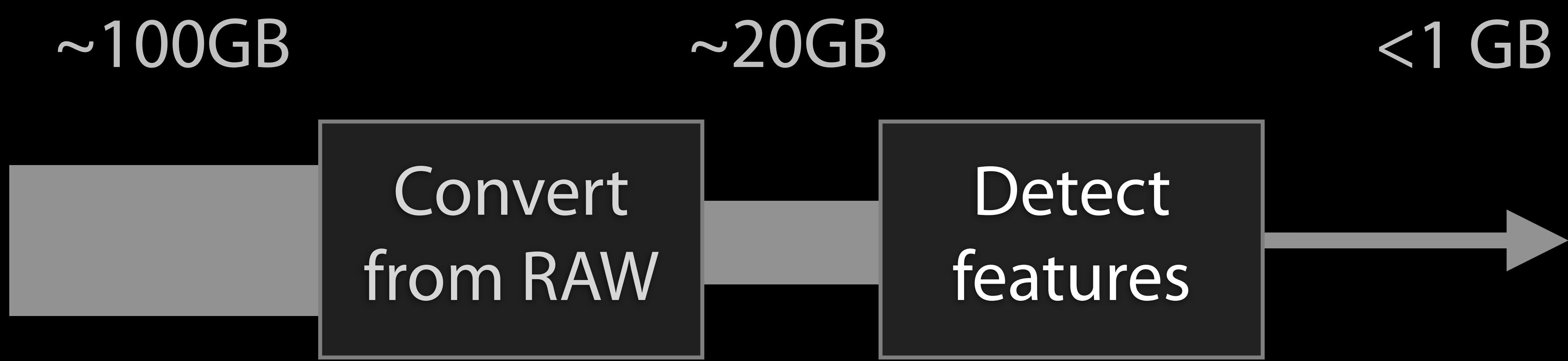
Sunday, August 1, 2010

Due to the nature of photographs created using our setup, detecting hairs corresponds to finding ridge features in the input photographs.
There are a number of ways to do this, and we chose to make modifications to the commonly used Canny edge detector with the purpose of turning it into a ridge finder.

We also use the scale-space framework to tune our filter so that it responds to features at exactly the width of a hair.

Because later stages of the pipeline greatly benefit from accurate 2D ridge orientation information, we convolve all input photographs with a filter bank of 32 7x7 Gabor filters and record the angle of maximum response separately for each pixel. Here are examples of such filters, and you can see that these will respond strongly to ridge features at various different angles.

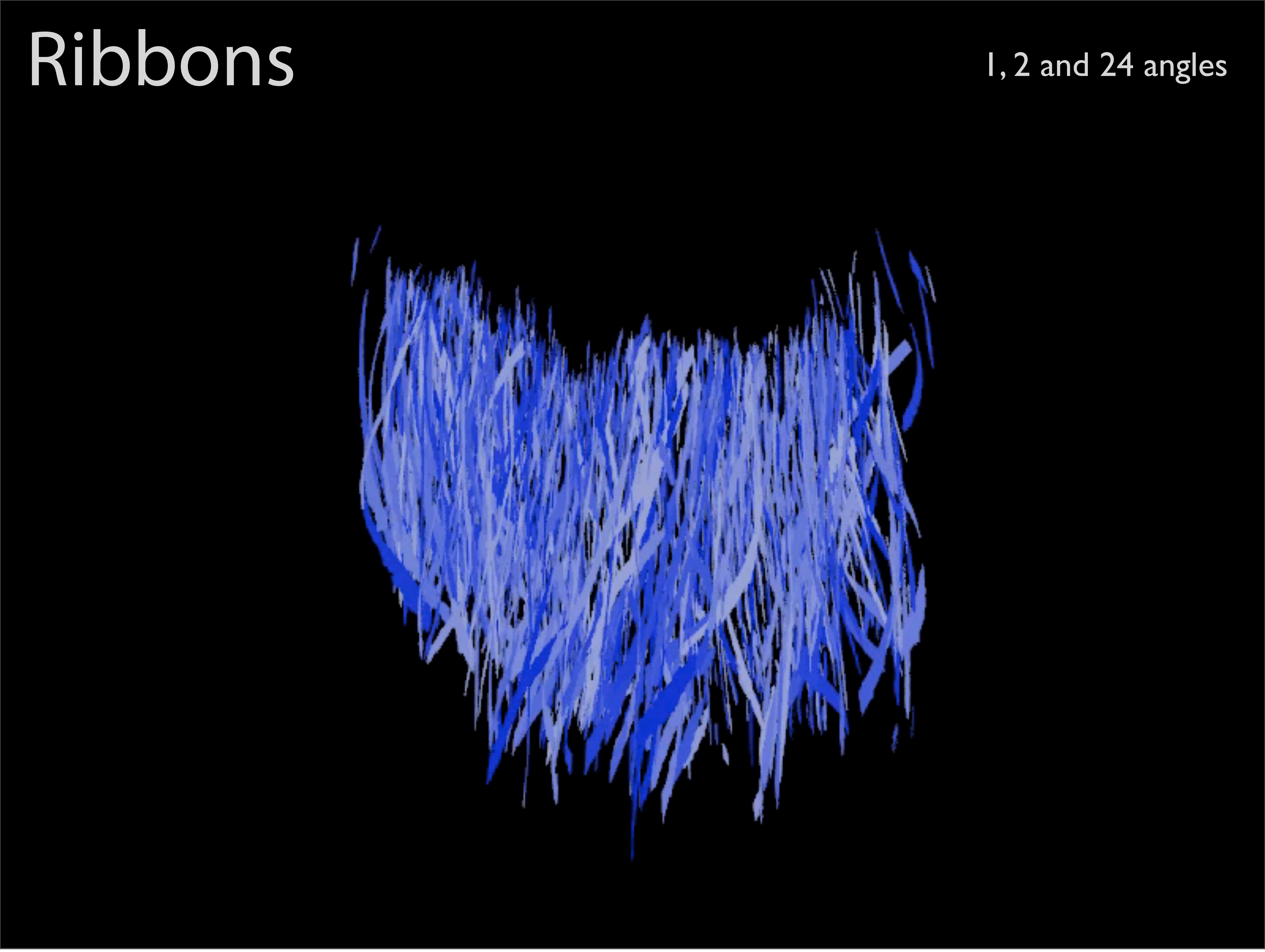
Sparsity



Filtering causes a significant reduction in the amount of data.

Sunday, August 1, 2010

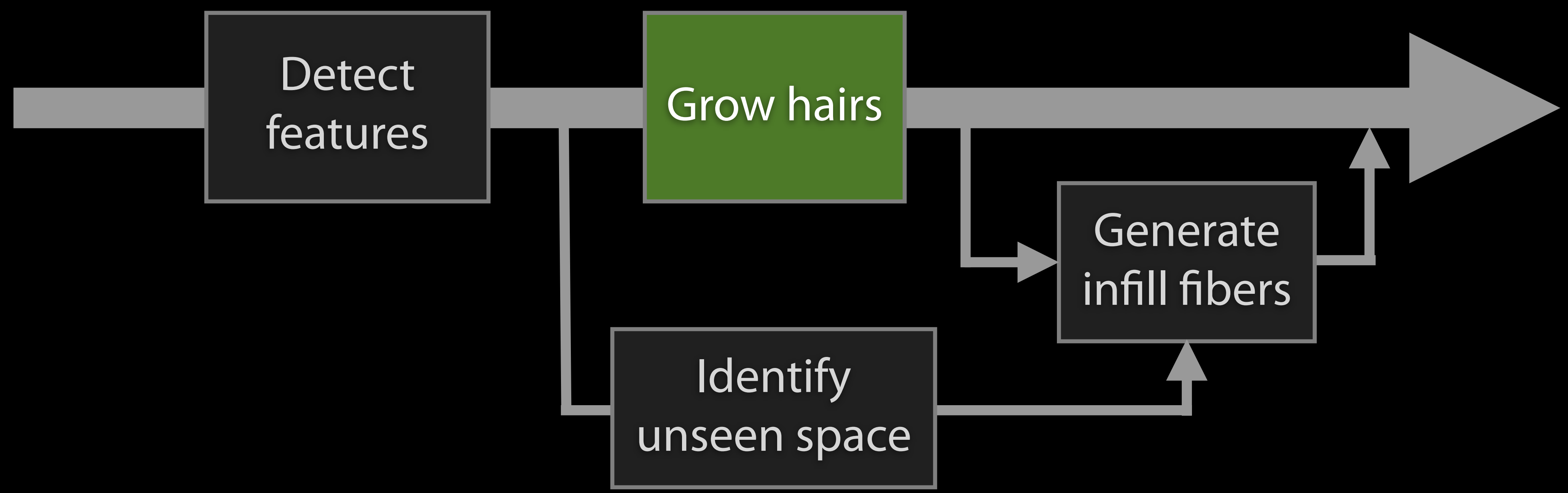
Another important feature of the ridge detection is the introduction of sparsity. The input photographs weigh in at about 100 gigabytes, which is quite cumbersome. After the filtering step, all data can be kept in main memory, which significantly accelerates the later processing stages. Next, I'll show the sweep we saw earlier, but now with the detected ridge features overlaid on top.



Sunday, August 1, 2010

The three-dimensional analog of the stars we saw previously are mutually intersecting ribbon-like shapes. Hairs are located at the common intersection of several such ribbons. Here, using a different hue for each angle, we show the detected features as seen from 1, 2 and 24 different rotation angles.

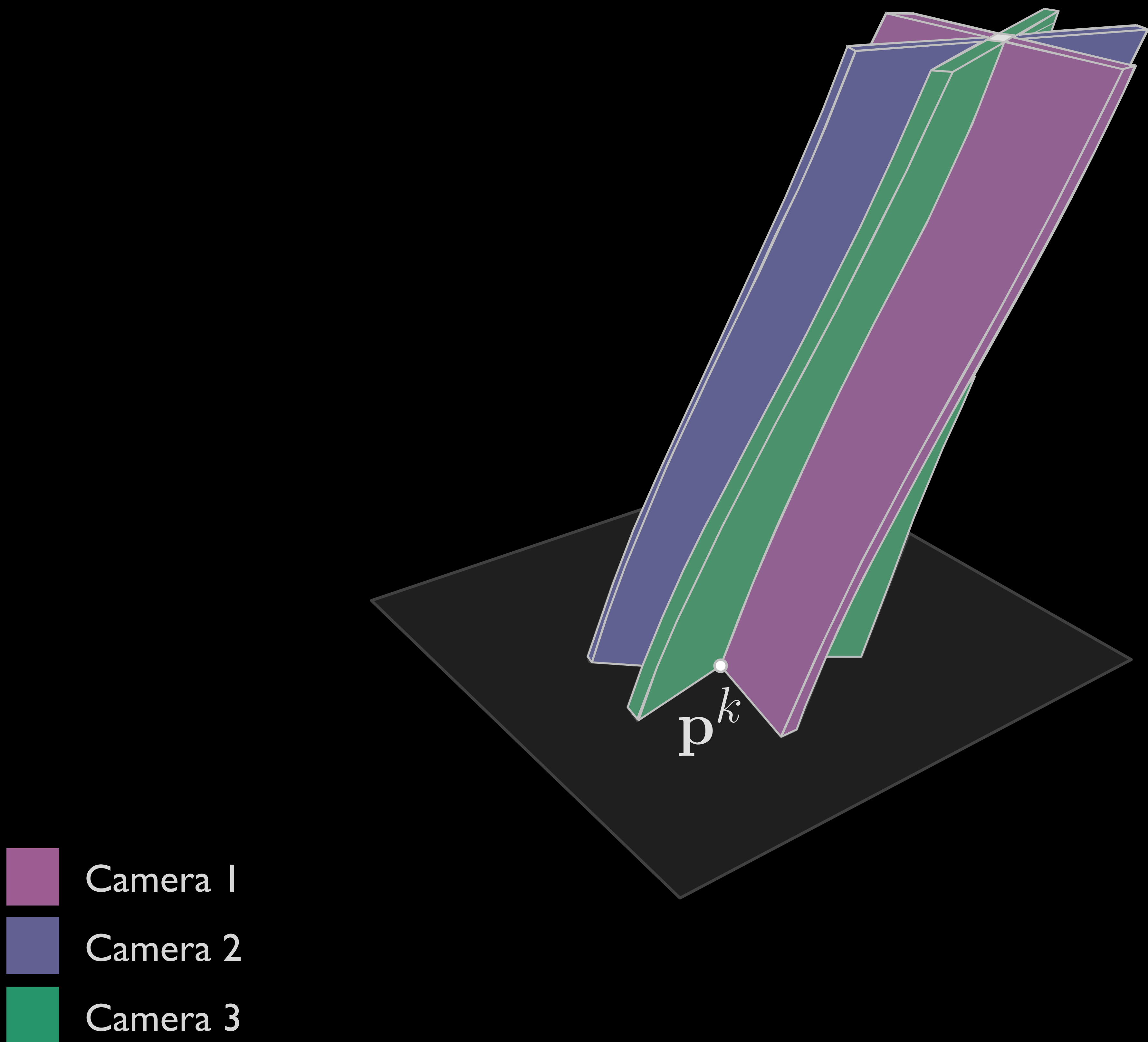
Pipeline overview



Sunday, August 1, 2010

With the filtering done, let's take a look at the fiber-growing stage.

Growing fibers

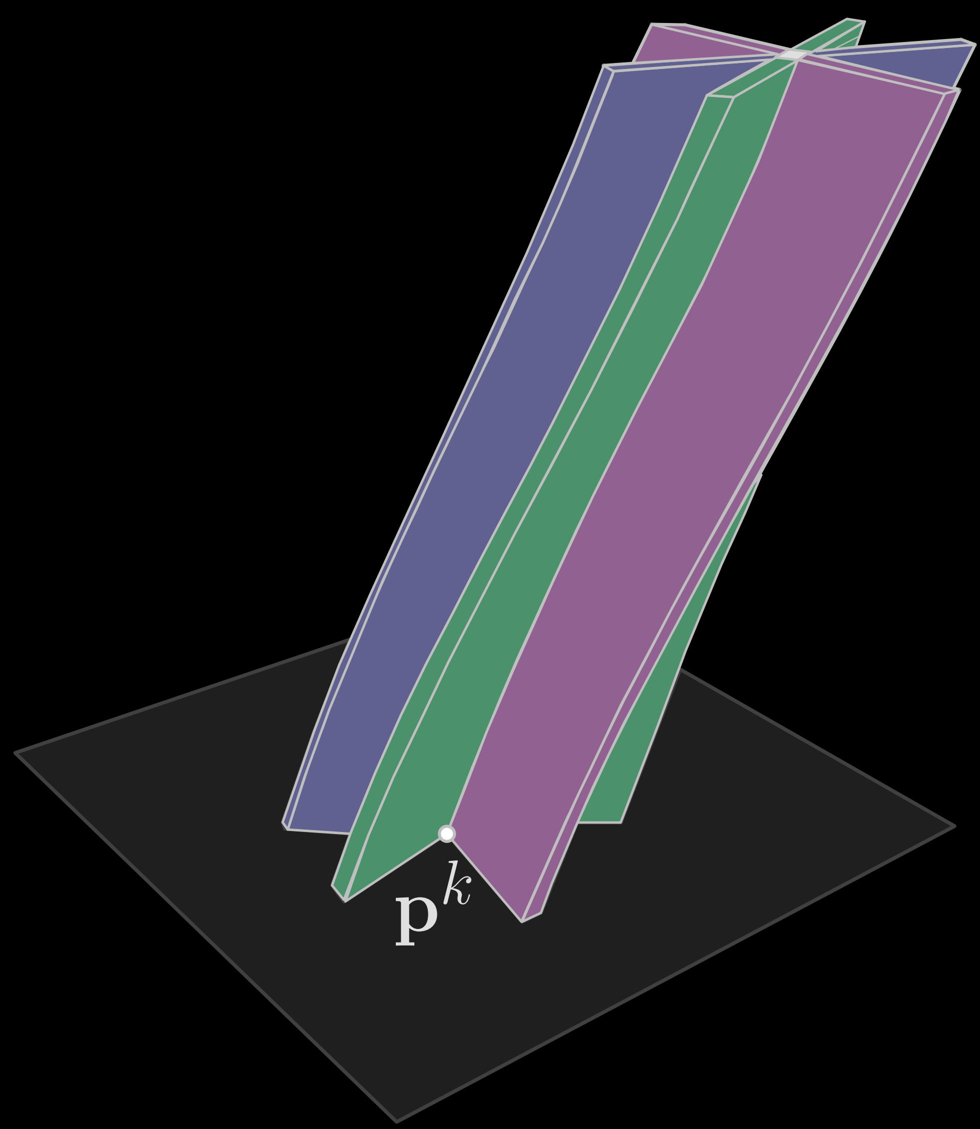


Sunday, August 1, 2010

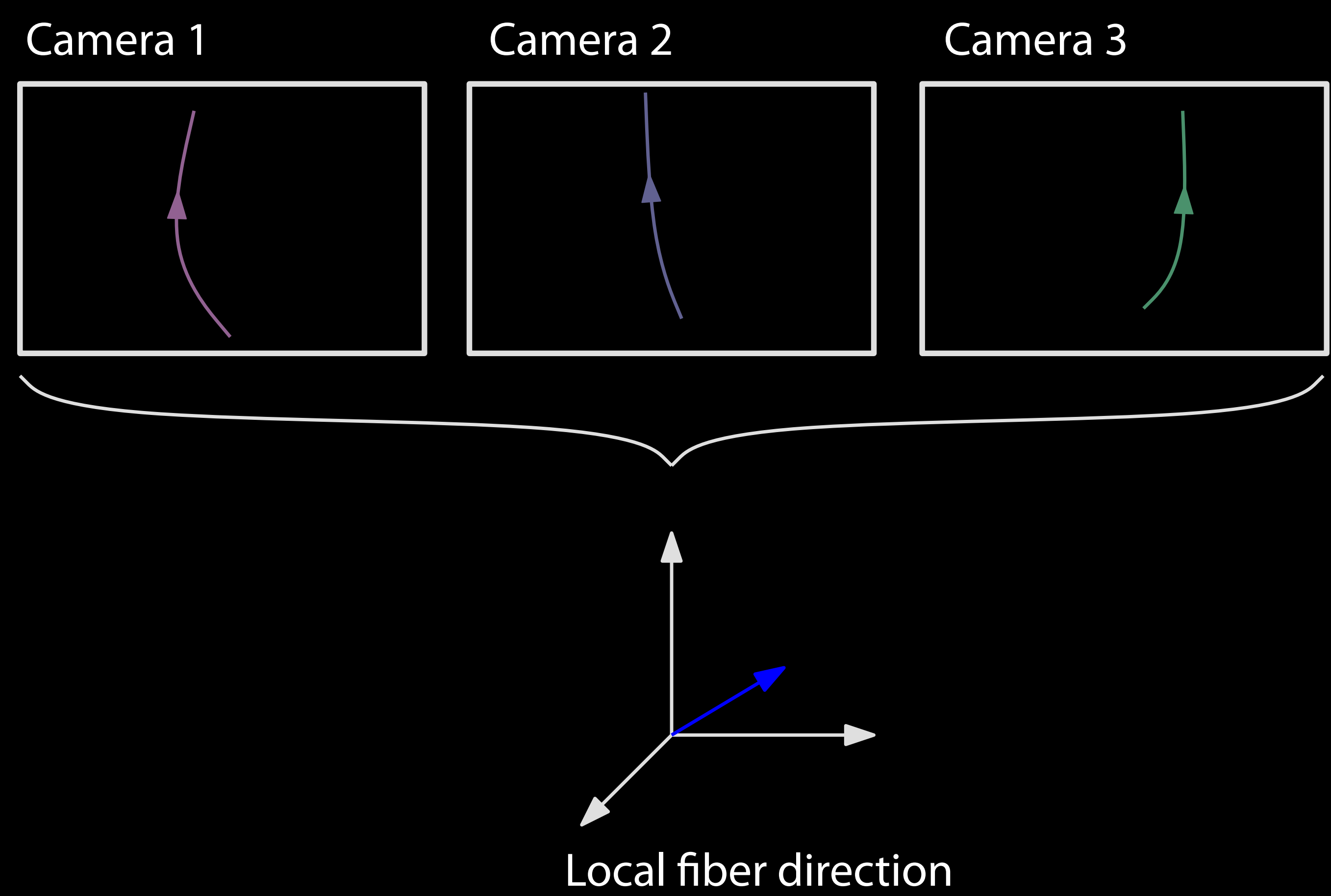
Here is an example of such a set of ribbons. The colors correspond to different camera angles, and the opaque regions represent detections of a hair that is effectively smeared out in depth according to the translation directions of the respective cameras.

Suppose we are given a position on the centerline of such a set of ribbons. To reconstruct the path of the associated hair, all we have to do is to follow the centerline. One piece of information, which makes this task significantly easier is that each voxel of a ribbon once originated from a detection in an input photograph -- which means that it has an associated 2D orientation.

Growing fibers



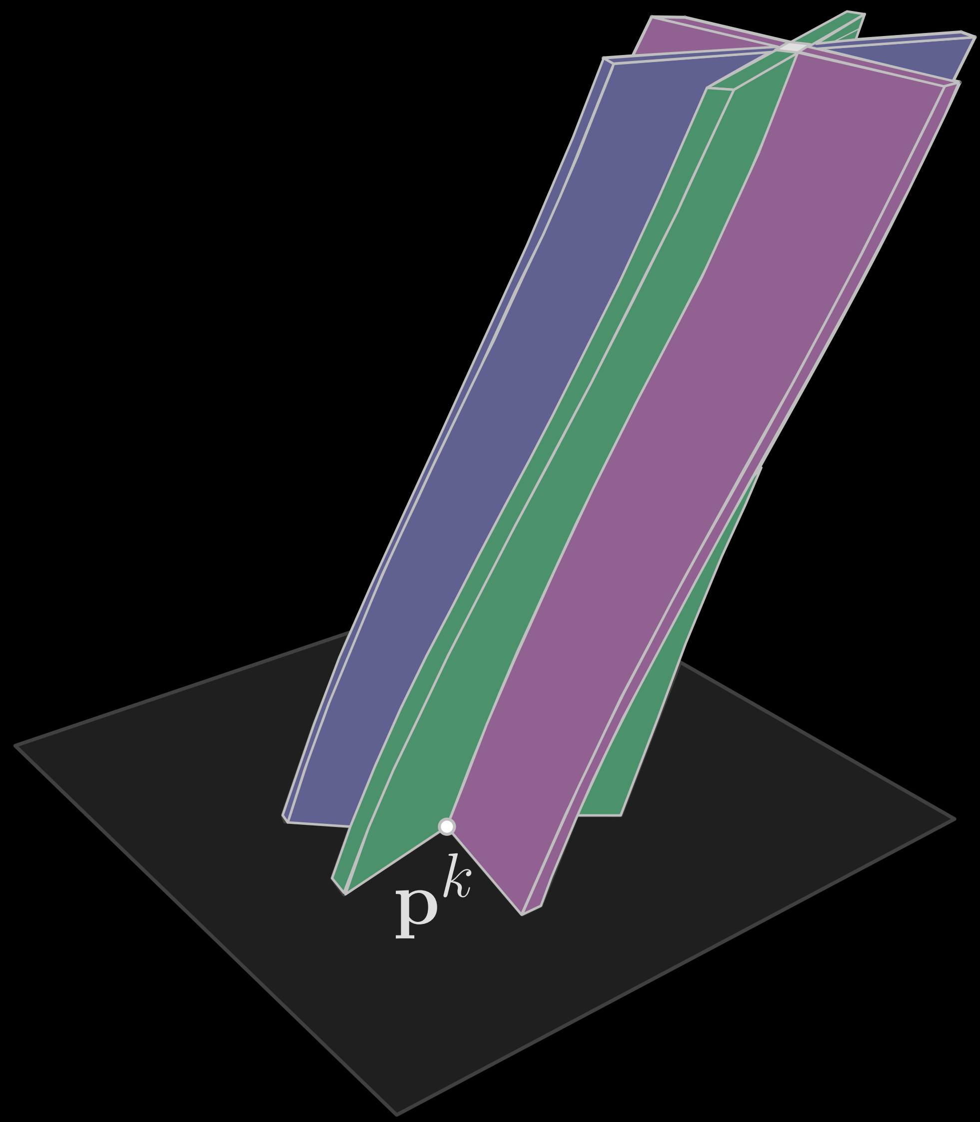
Goal: find a 3D direction that is compatible with its 2D projections



Camera 1
Camera 2
Camera 3

So the problem we need to solve is: given a number of 2D orientations observed from different cameras, find a 3D direction that is consistent with all of them.

Growing fibers



This is an SVD problem!

$$\begin{bmatrix} - & n_1 & - \\ - & n_2 & - \\ - & n_3 & - \end{bmatrix} = \underbrace{\begin{bmatrix} | & | & | \\ u_1 & u_2 & u_3 \\ | & | & | \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} - & v_1 & - \\ - & v_2 & - \\ - & v_3 & - \end{bmatrix}}_{V^T}$$

3rd right singular vector

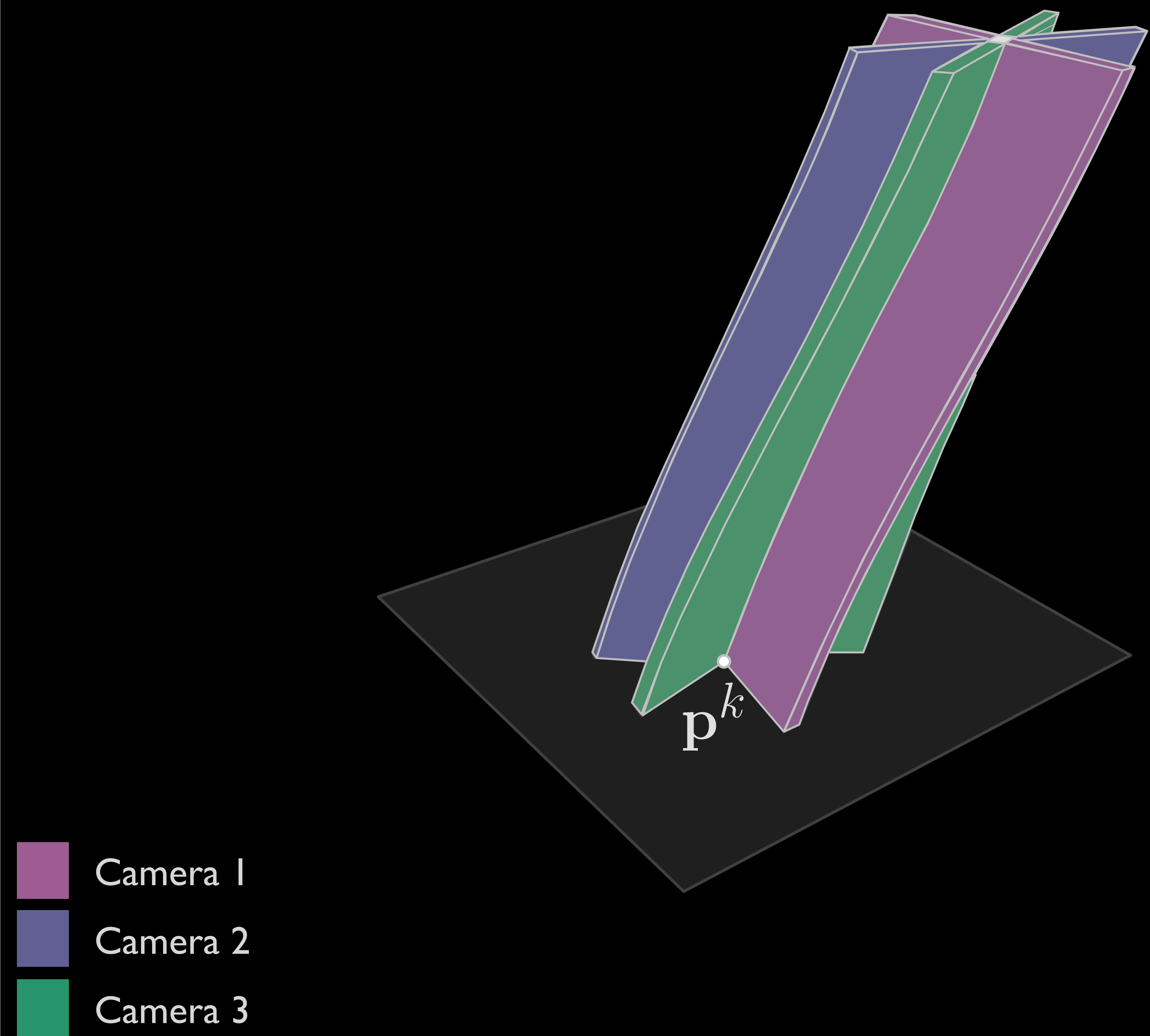
Camera 1

Camera 2

Camera 3

This is an SVD problem, and the least-squares solution can be found by searching for the direction, which is most orthogonal to the normals of these ribbons.

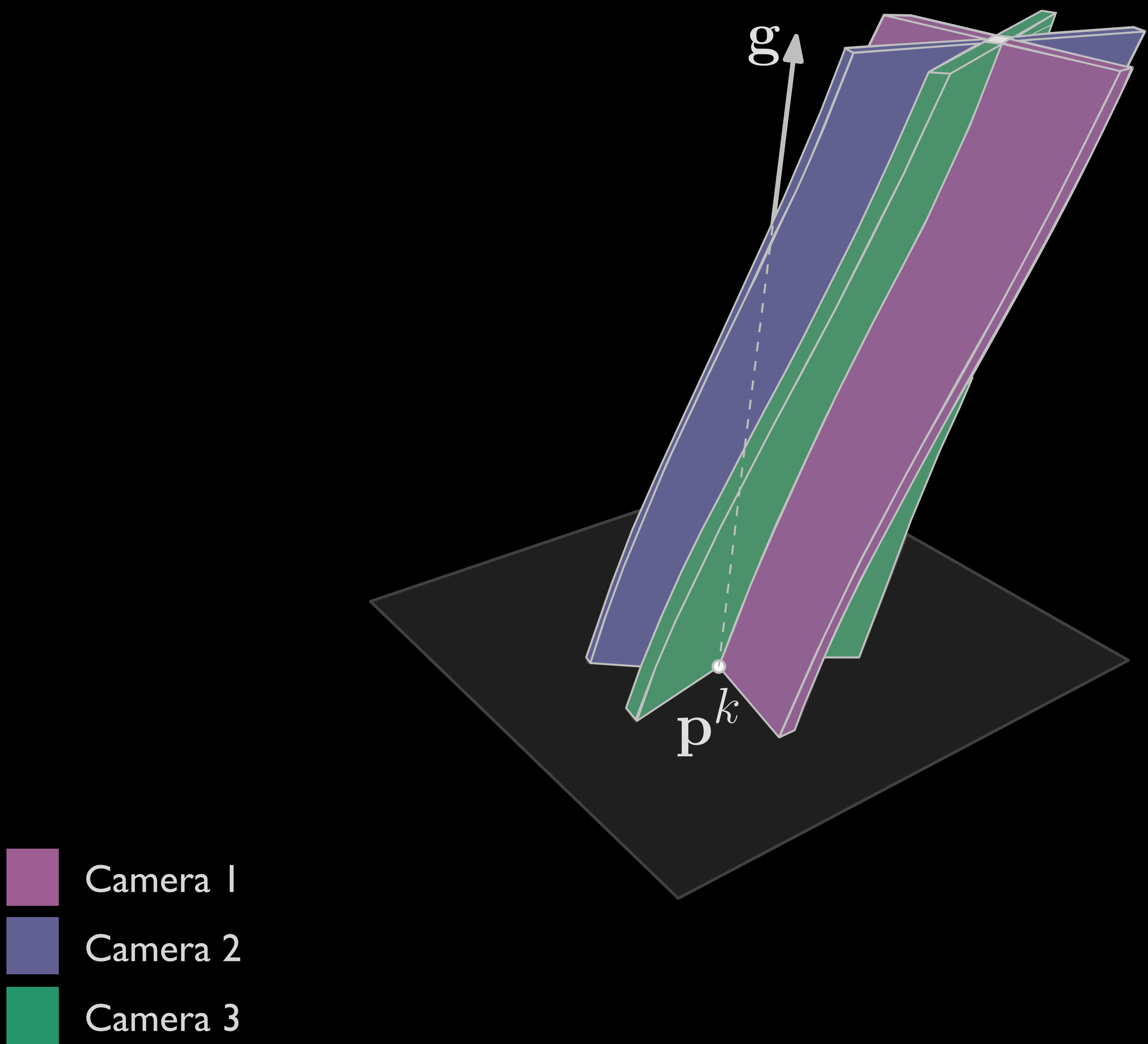
Growing fibers



Sunday, August 1, 2010

Suppose now that we are able to compute the fiber tangent direction in this way. A simple algorithm would be to compute the tangent, take a step in that direction, and to repeat this process until the end is reached.

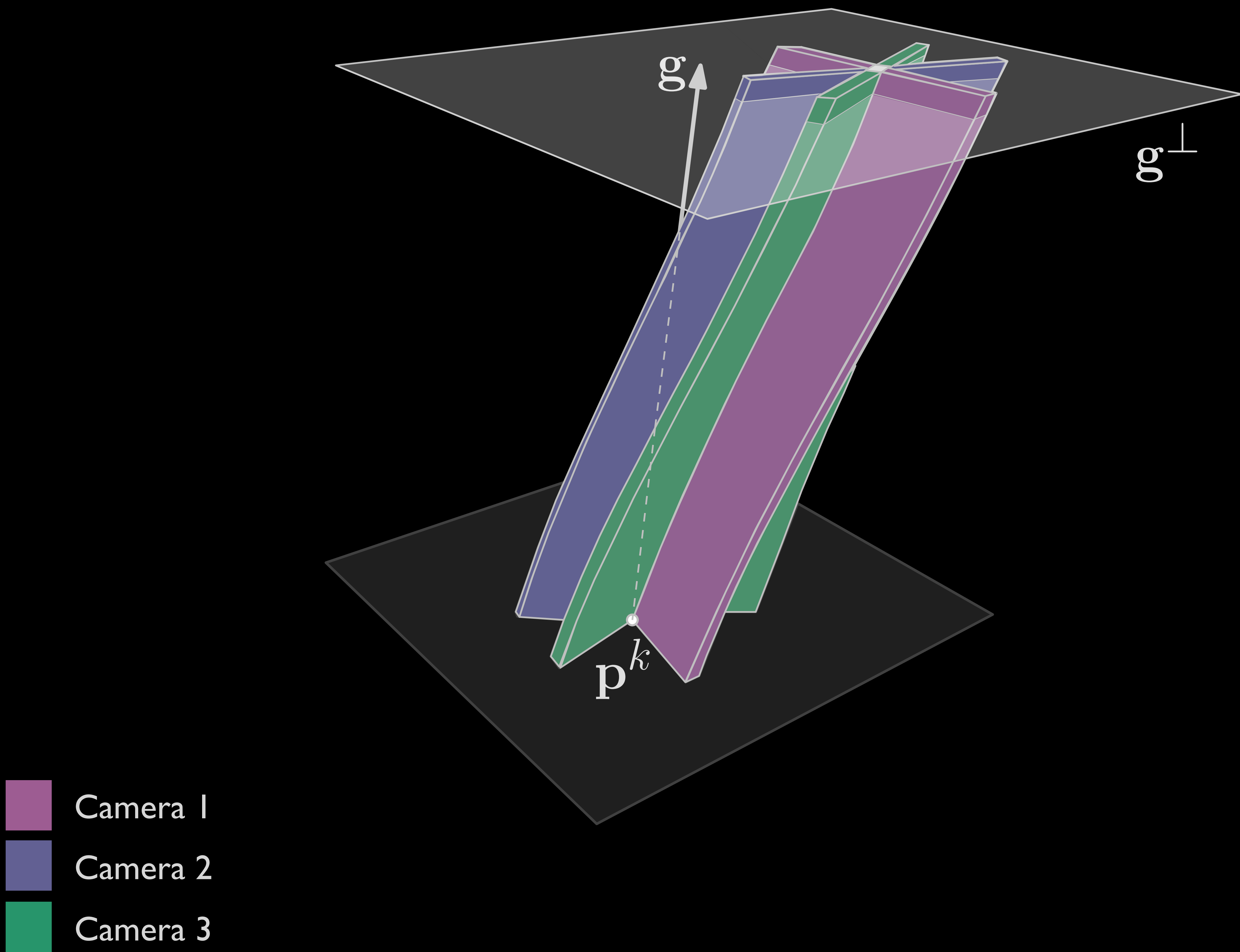
Growing fibers



Sunday, August 1, 2010

In a perfect world, this approach might actually work quite well. Unfortunately, there are some problems with it in practice. Mainly due to noise and error accumulation, we'll eventually stray away from the centerline, at which point we can't accurately estimate the tangent direction anymore. In this example figure, we of course massively overshoot due to a very large step size. But this problem persists even when the step size is chosen to be very small.

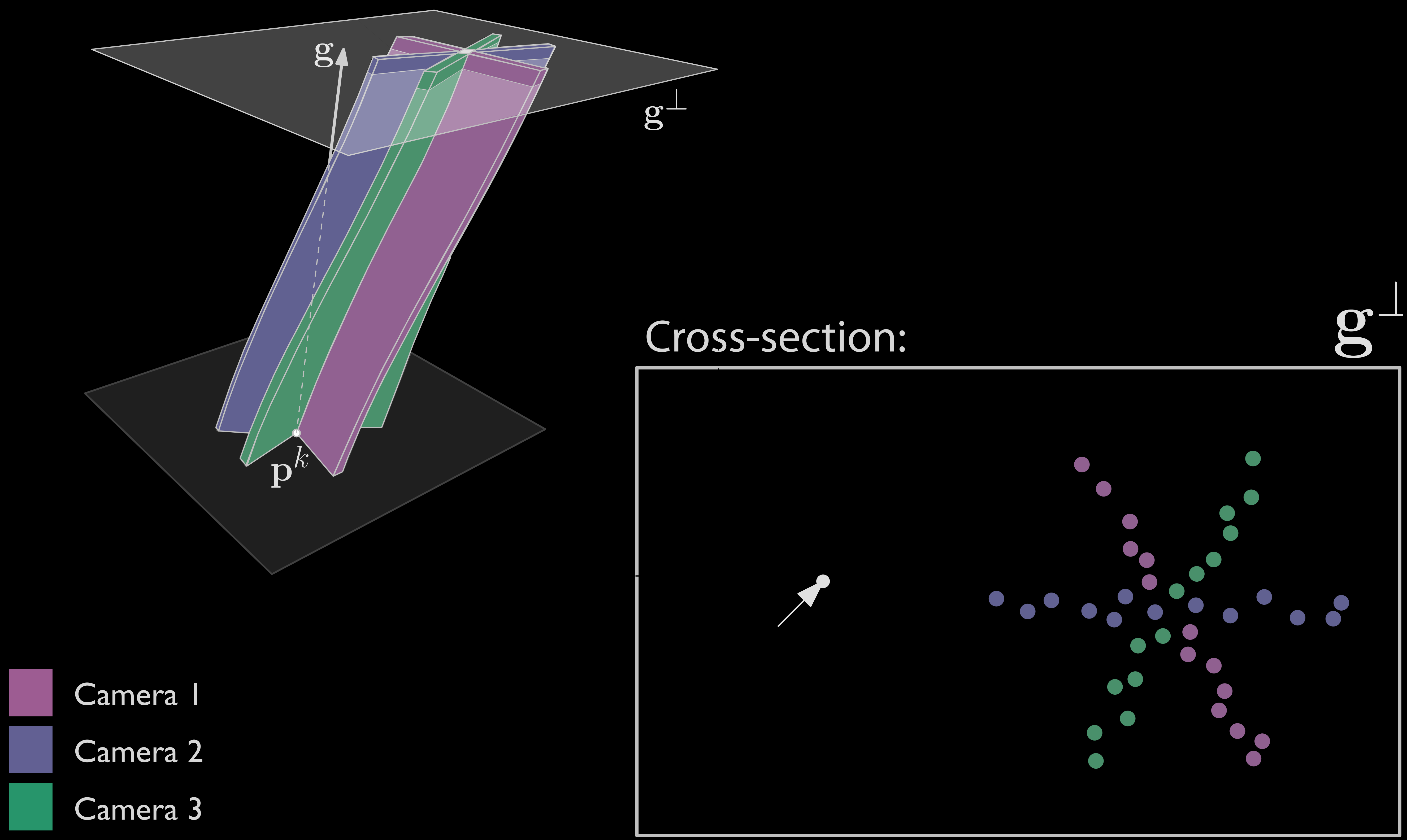
Growing fibers



Our solution is to perform a re-centering correction after each step, which ensures that the next position snaps back on the centerline.

To do this, we take a look at the space g-perp that is orthogonal to the growing direction and we extract a cross-sectional slice around the proposed next position.

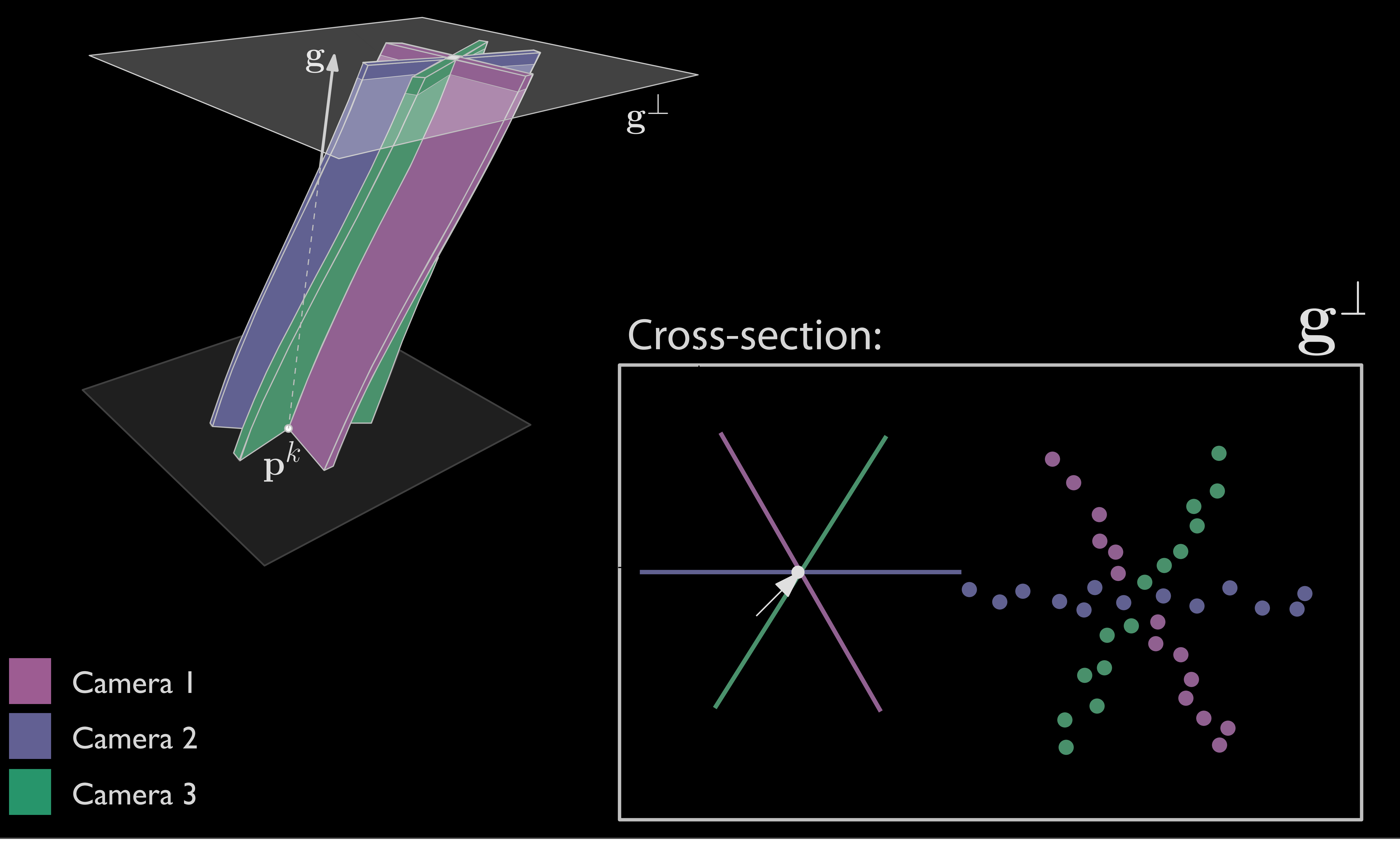
Growing fibers



When looking at the points lying in the cross-section, we might get something like this: a number of hair observation, which are arranged in a star-like shape, modulo noise, and a proposed position which has moved away from the center.

Here, it is useful to remember that we’re tracking hairs (and our algorithm does this using very small steps on the order of fractions of a millimeter). On this scale, hairs are essentially straight lines, and therefore, the local tangent direction changes only very little from step to step.

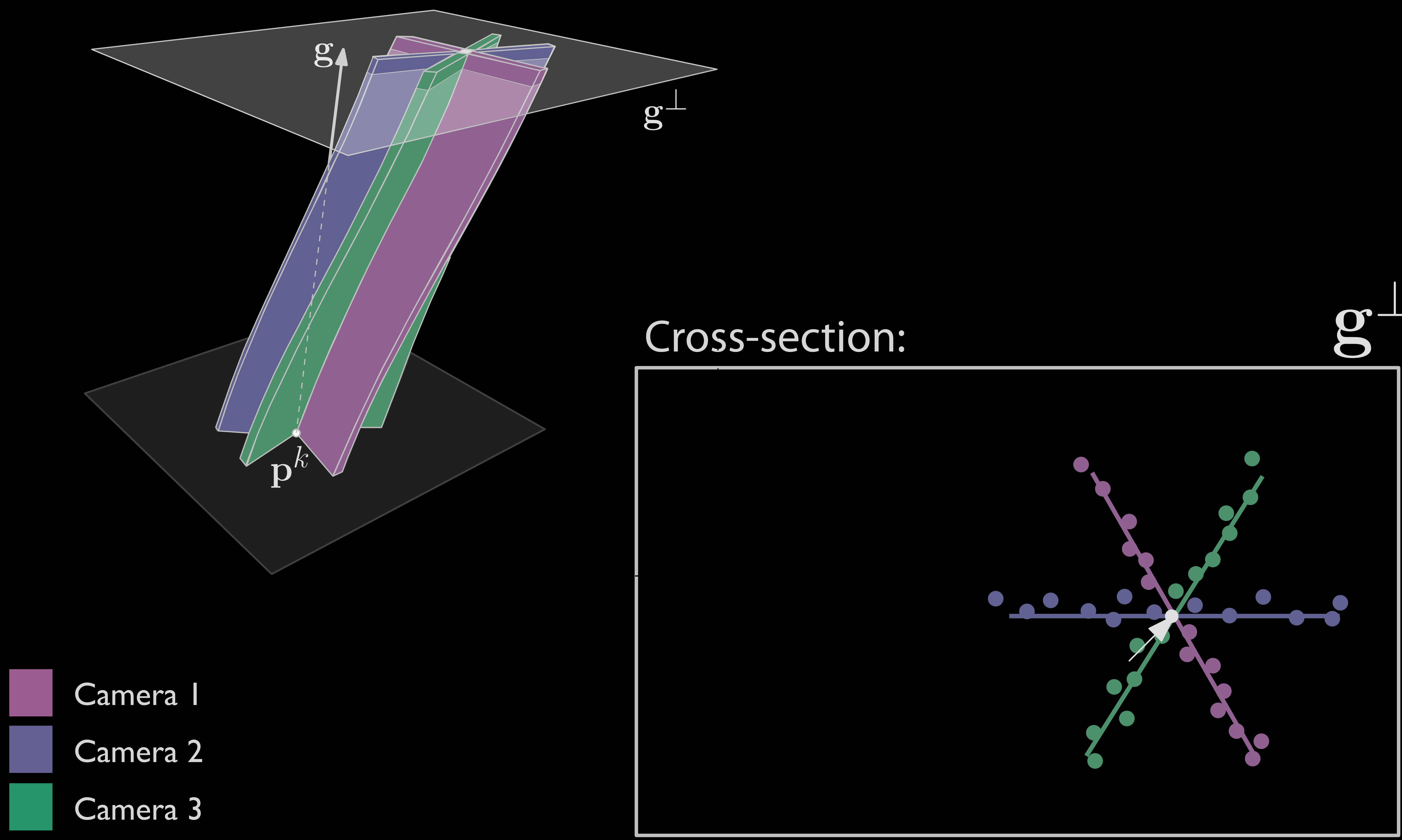
Growing fibers



Sunday, August 1, 2010

With this piece of information at hand, it is possible to compute the star-shaped stencil corresponding to the expected observation positions of a hair with that direction.

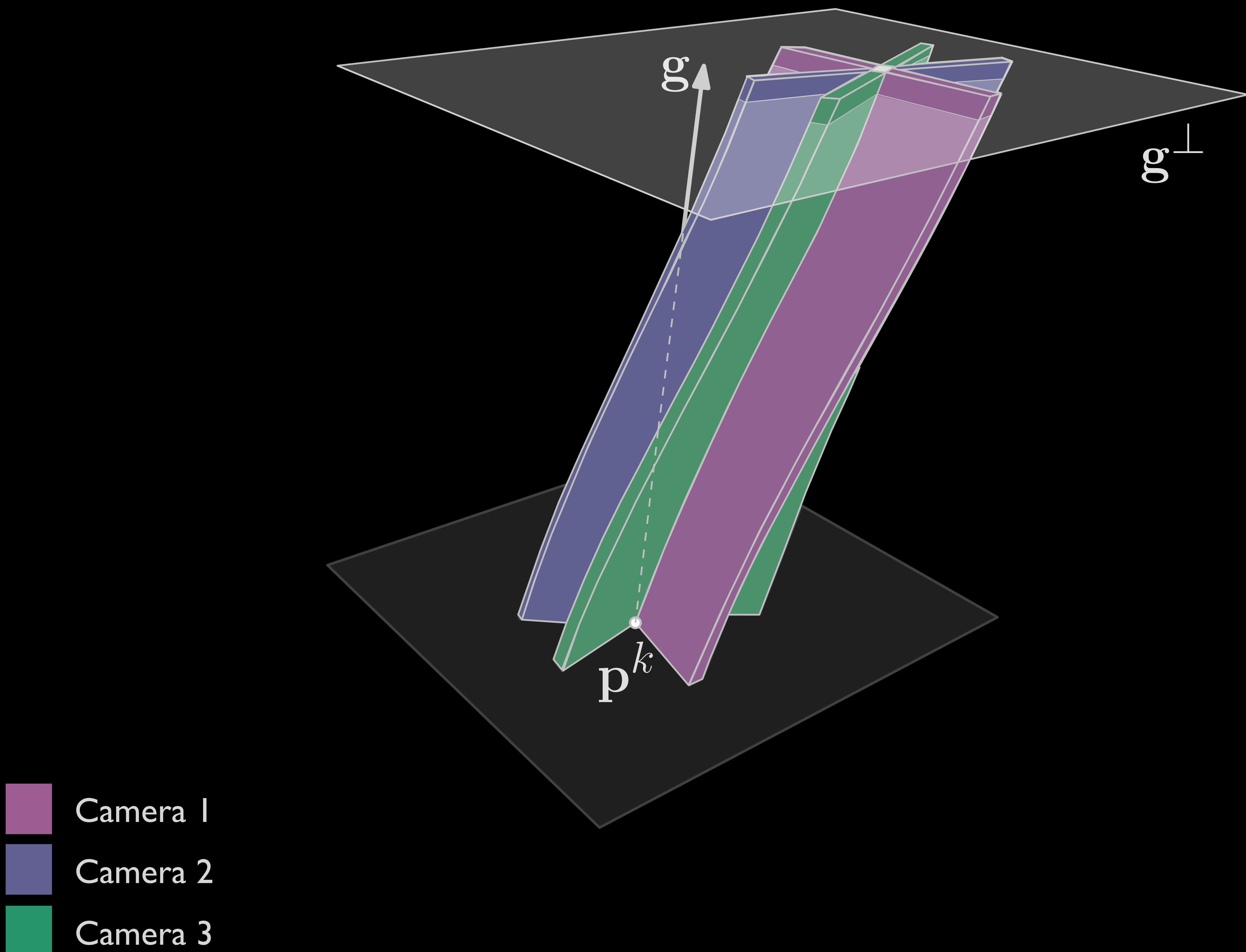
Growing fibers



Sunday, August 1, 2010

To perform the re-centering correction, all that needs to be done is to align the stencil with the actual observations. This can be expressed as as series of simultaneous point-to-plane problems, which results in a linear least squares problem.

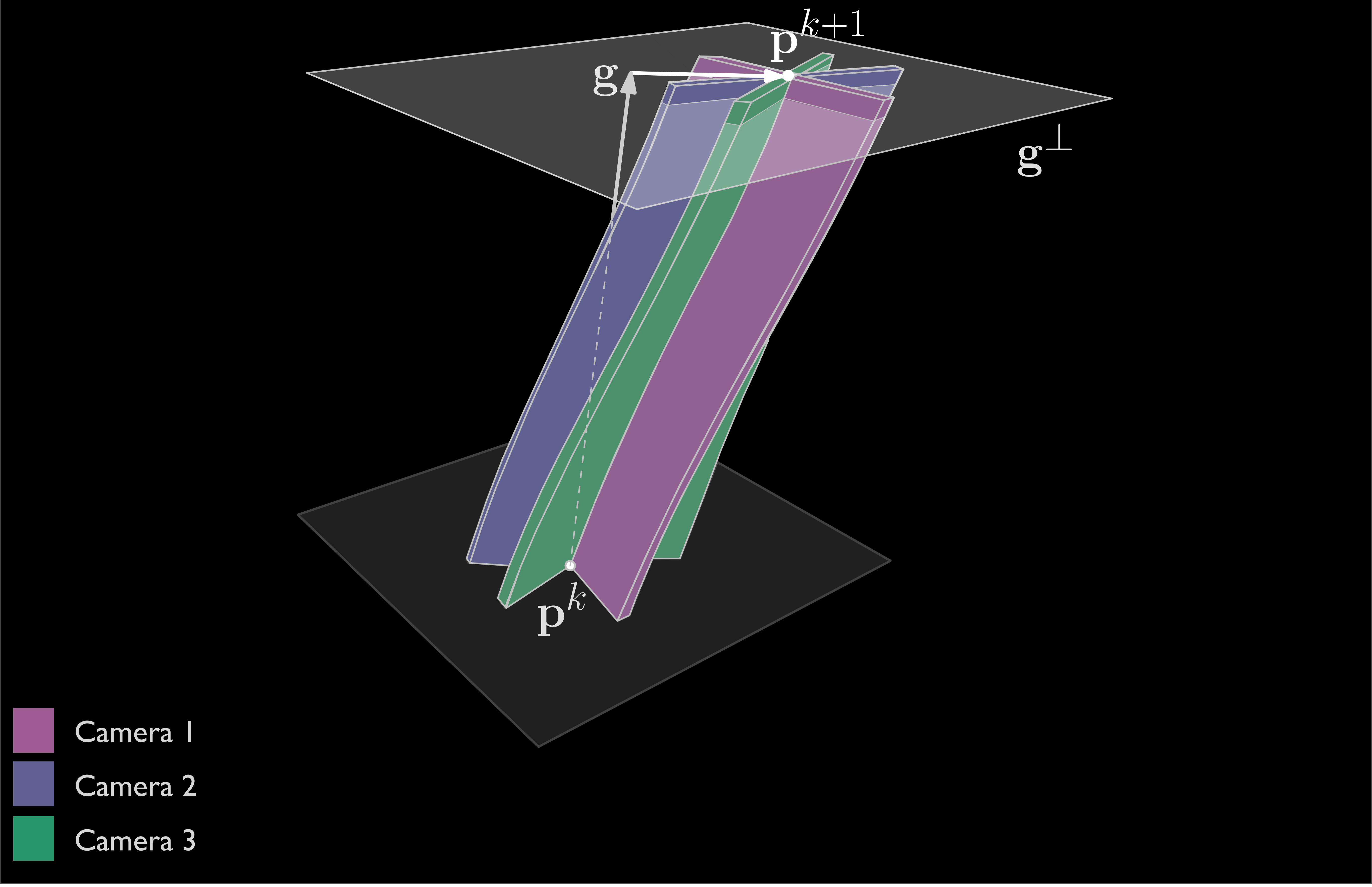
Growing fibers



Sunday, August 1, 2010

Finally, once we know the correction in g-perp, we get the starting position of the next iteration, and the process repeats.

Growing fibers



Sunday, August 1, 2010

After putting all of this together, we end up with a simple iterative algorithm that solves alternating least squares problems.

Algorithm pseudocode

```
GROW-HAIR()
p ← GENERATE-SEED()
repeat
    if FIND-ORIENTATIONS() < 3
        break
    d ← SOLVE-DIRECTION()
    p ← p + h d
    p ← p + SOLVE-CORRECTION()
REMOVE-FEATURES()
```

Sunday, August 1, 2010

Here is a pseudocode version of this algorithm. Crucial to the numerical stability of these least squares problems is that that there are enough observations, and therefore the growing process stops when this number falls below a certain threshold.

Algorithm pseudocode

```
GROW-HAIR()
p ← GENERATE-SEED()
repeat
    if FIND-ORIENTATIONS() < 3
        break
    d ← SOLVE-DIRECTION()
    p ← p + h d
    p ← p + SOLVE-CORRECTION()
REMOVE-FEATURES()
```

Sunday, August 1, 2010

Other than that, the iteration proceeds exactly as we talked about before: A direction is estimated, then we take a step, and finally a correction is added.

There are two pieces left, which I haven't talked about:

Algorithm pseudocode

```
GROW-HAIR()
  p ← GENERATE-SEED()
  repeat
      if FIND-ORIENTATIONS() < 3
              break
      d ← SOLVE-DIRECTION()
      p ← p + h d
      p ← p + SOLVE-CORRECTION()
  REMOVE-FEATURES()
```

Sunday, August 1, 2010

Other than that, the iteration proceeds exactly as we talked about before: A direction is estimated, then we take a step, and finally a correction is added.

There are two pieces left, which I haven’t talked about:

Algorithm pseudocode

```
GROW-HAIR()
p ← GENERATE-SEED()
repeat
    if FIND-ORIENTATIONS() < 3
        break
    d ← SOLVE-DIRECTION()
    p ← p + h d
    p ← p + SOLVE-CORRECTION()
REMOVE-FEATURES()
```

Sunday, August 1, 2010

Other than that, the iteration proceeds exactly as we talked about before: A direction is estimated, then we take a step, and finally a correction is added.

There are two pieces left, which I haven't talked about:

Algorithm pseudocode

```
GROW-HAIR()
p ← GENERATE-SEED()
repeat
    if FIND-ORIENTATIONS() < 3
            break
    d ← SOLVE-DIRECTION()
    p ← p + h d
    p ← p + SOLVE-CORRECTION()
REMOVE-FEATURES()
```

Sunday, August 1, 2010

First of all, I never said where the initial position comes from. We use a simple heuristic, which works well in practice. The idea is to look for positions where many observations from different views intersect. This can be taken as evidence of a hair passing through the position.

Algorithm pseudocode

```
GROW-HAIR()
p ← GENERATE-SEED()
repeat
    if FIND-ORIENTATIONS() < 3
            break
    d ← SOLVE-DIRECTION()
    p ← p + h d
    p ← p + SOLVE-CORRECTION()
REMOVE-FEATURES()
```

Sunday, August 1, 2010

Secondly, after a hair has been grown from top to bottom, we want to avoid detecting that same hair in the future. All of the features used during the growth process are therefore removed to ensure this.

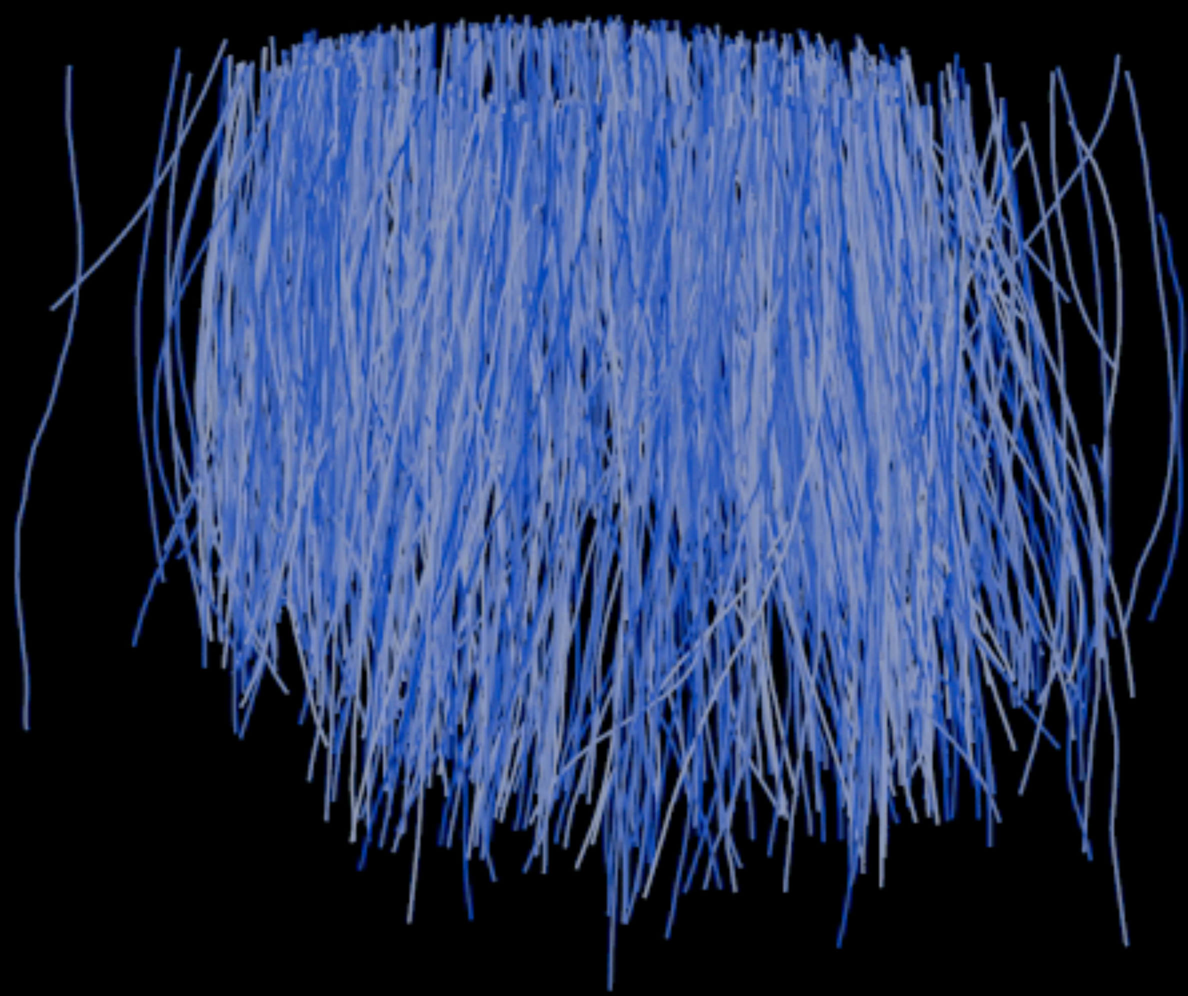
Note that all steps that access the feature data can be implemented using kd-trees, which makes it possible to process a whole dataset in just a few minutes.

Demo

Sunday, August 1, 2010

Here is a visual demonstration of the growing algorithm at work. On the right side, you can see two different cameras tracking the current growth position. The little moving box is the region where the algorithm searches for seed points. When no more can be found, it advances by one position. (This demonstration was massively slowed down so that it can be watched more easily).

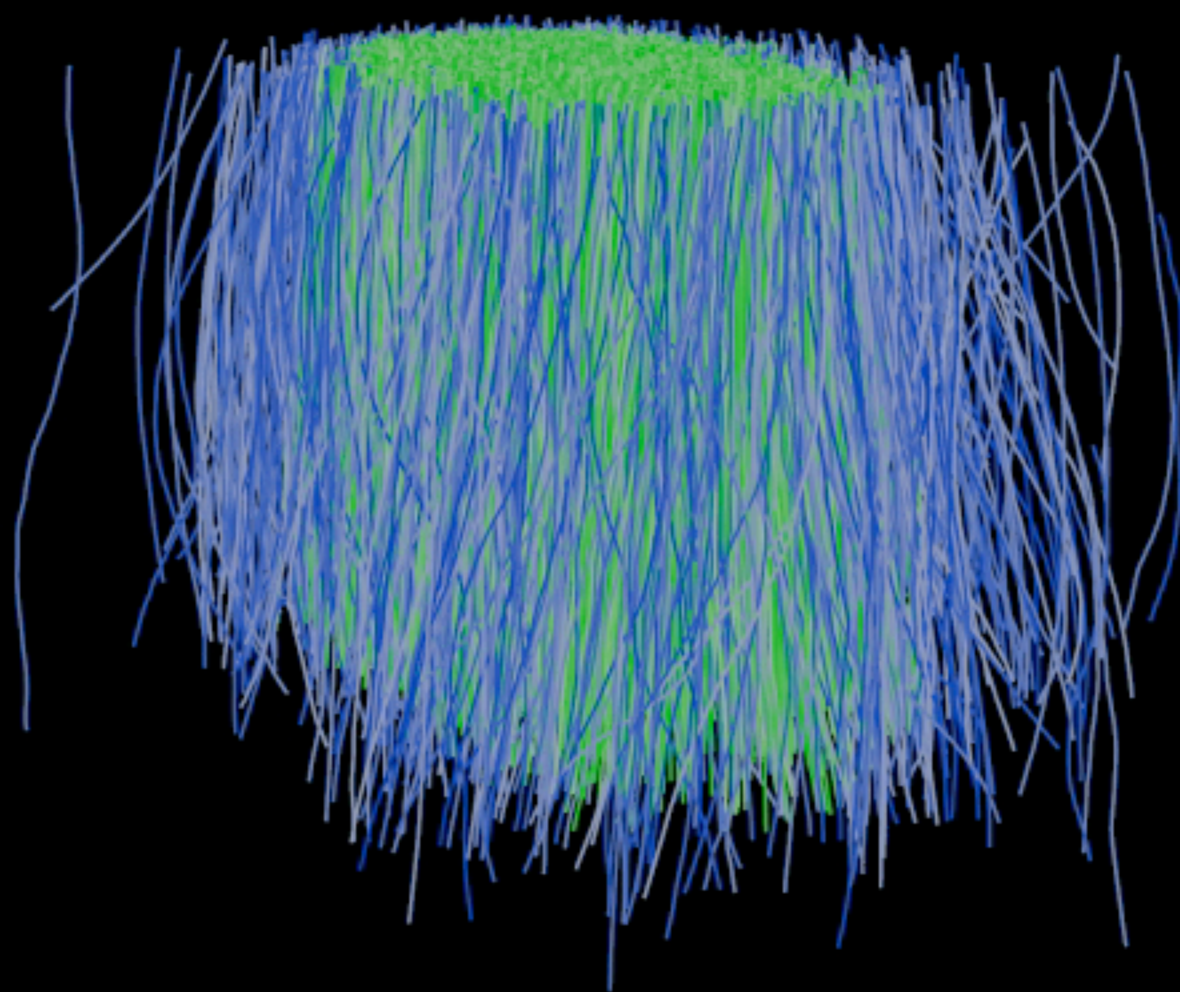
Infill generation



Sunday, August 1, 2010

The resulting set of curves captures most of the hair visible from the outside of the hair assembly. But there are many hairs on the inside that cannot be recovered due to the large amount of occlusion. For some applications, that might not be of much concern -- though if the goal is to produce renderable geometry, the empty regions must be filled.

Infill generation



Sunday, August 1, 2010

The occluded fibers fortunately have a limited influence on the appearance of the hair, which lets us fill these areas using generated fibers. As I said earlier, the paper contains information on an automated workflow to do this.

Timings

Filtering	Fiber-growing	Infill generation
6 hours	5 min	negligible

(On a 4-core Intel Core i7 940)

The single most time-consuming part is the filtering stage, with most time being taken up by the Gabor filter convolutions.

The paper actually lists 6hrs for the fiber-growing phase, but this can be reduced to 5 minutes when implemented with the proper acceleration structure. The source code released with the paper implements this version.

Infill generation takes less than a minute.

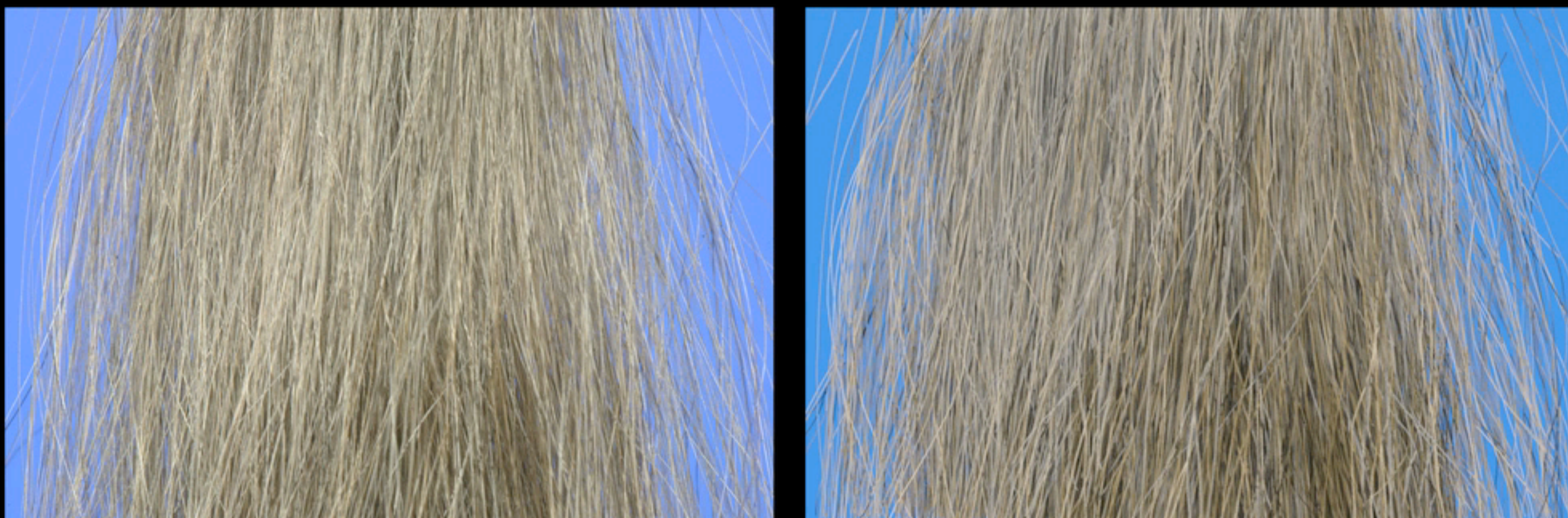
Validation

Two comparisons:

- *straight*: brown hair, combed straight
- *wavy*: blond hair, slightly curled with a curling iron

We now show two comparisons of captured video footage against the reconstructed geometry rendered using physically-based methods.

Validation



video rendering

Note that almost all of the visible hair fibers have been reconstructed. Also, the infill fibers are mostly hidden, but they produce appropriate opacity and background color.

Conclusion

- Capturing the geometry of a large fraction of the visible fibers is feasible
- Realistic 3D fiber arrangement enables ground-truth comparisons
- Achieved realistic results without image-based rendering

Code and datasets:
http://www.cs.cornell.edu/projects/HairCapture

In conclusion, we have demonstrated an algorithm that captures a large fraction of the visible fibers in a hair sample using a series of sweeps at a shallow depth of field.
The resulting models have realistic fiber arrangements, which enables ground-truth comparisons using physically-based rendering methods.
Our reconstructions provide the first views we know of into the actual 3D arrangement of fibers in real hair assemblies, and we hope that they will be useful in the development of future modeling tools and in the verification of hair rendering algorithms. The code and data are both available for download and I encourage you to play with them.

Acknowledgements

- NSF Grants CCF-0347303, CCF-0541105
- Unilever Corporation
- Noah Snavely and Robert Velthuizen

Code and datasets:
http://www.cs.cornell.edu/projects/HairCapture

We would like to thank our funding agencies and our colleagues for helpful conversations.