Transparent material/fluid scanning
    -tying into tomography
        -contribute to science but have fun (toy aspect- new scan acquisition i.e. Jellyfish)
-computer graphics/science acquisition
-making things invisible to the naked eye visible...
-computational photography (new) vs tomography (old)
-different problems/applications
    -tied together with tomography in 3D reconstructions

-issues when doing tomography
    -given volume, volumetric (scalar) property (on volume)
-take a number of observations (line integrals)
    -take enough observations to be able to solve for any point in space on the volume
    -we know the ray geometry path s
-xray straight through tissue (ray geometry)
-volume in liquid with same refractive index to keep rays straight
-(ray geometry has continuous refractions)

Schlieren tomography

Problems:
-what are good camera setups? Etc...
-with glass (rigid body) can take as many pictures as desired (geometry doesn't change: unlimited time)
-fluid and jellyfish change and move (all pictures must be taken simultaneously)
    -camera array needed: good configuration?
    -bad to have all cameras on one side (ideally 180 degree setup)
    -make sense to have vertical parallax as well as horixzontal one?

Once the setup is obtained:
-must solve tomography problem (linear, matrix discretize)
-undetermined system and noise (need regularization)

Visual Hull Constraints
-intersection of multiple silhouettes
-removes degrees of freedom and regularizes
-additional kinds of regularizations?

Inverse relations
-deconvolve sparse gradient
    -smooth almost everywhere except change of material
-binary material?
    -surrounding medium and object scanned
-problem specific

Other possibilities
-preconditionals
-solvers

Problem:
-matrix is huge (256^3, 24M voxels, 24Mx24M matrix...)
-high quality reconstruction in every slice?
    -cubic reconstruction?

Work arounds:
SART (simultaneous algebraic reconstruction technique)
-used for glass scanning (paper available)
-volume rendering step
-residual back-projection step
    -initial estimate of volume (i.e. 0 or 1...)

    - volume render and get image
    - compare value from camera
    - project back and uniformly increment or decrement along ray (reduce residual)

- works on numerical analysis point of view (not efficient)
- conjugate gradient converges faster
- can use some kind of dual operators:
    - volume rendering: Ax voxel -> pixel
    - back projection: A^Tx pixel -> voxel
- instead of rendering matrix can generate matrix multiples

- preconditioners increase convergence
    - multi-grid preconditioners
- how to solve this without storing matrix
    - can use any matrix-free solver
    - reformulate as algebra problem
- cannot assume will (be able to) store matrix
- can do adaptive stuff (e.g. Non-uniform density structure)
- rigid reconstruction of one volume

- solve as basic optimization problem?
    - more general optimization problem

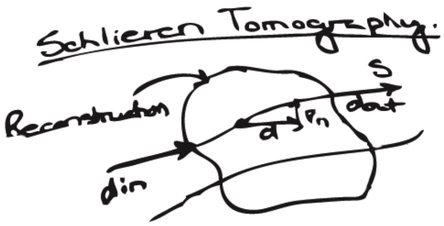
MRI/CT scan?

- x-ray sources orthographic
    - Fourier slice reconstruction
- no regularization in medicine
    - CT artifact vs tumor...

Problem specific stuff:

- vector (free scalars) components of refractive index gradient (want refractive index field)
- continuous vs discontinuous
    - problems with refraction and reflection

- change of direction is made by adding the gradient (of refractive index field) to the current direction



$$\frac{d\vec{d}}{ds} = \nabla n$$

- for more detailed information, see Brad's gas capture paper

- can integrate:

$$\vec{d}^{out} - \vec{d}^{in} = \int \nabla n \, ds$$

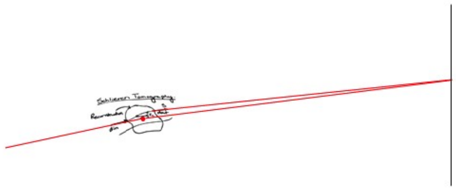
(known) (unknown-can solve for integral property)

d^{in}:

- camera callibration
- ray with respect to volume
- volume with every pixel in camera

d^{out}:

-harder (need some approximation)
-object assumed small relative to background/distance to background



<- (high frequency background)

-path from “center” of object traced to background
-difference between this and actual out path very small
-distance of object from background should be relatively large

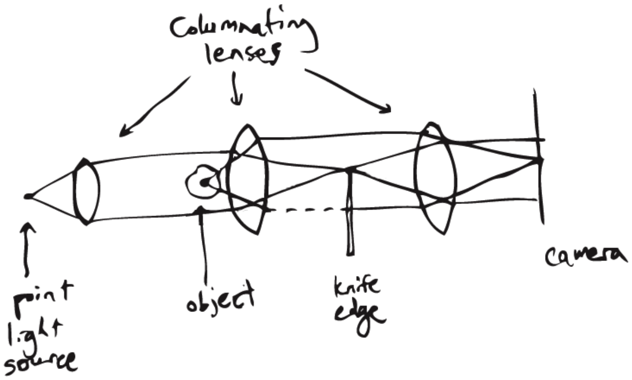
environment mapping
-mapping background to pixels
-need both background and object to be in focus
    -solution: move camera back

Background:
-BOS
-stripe patterns
-pixel to pixel mapping
-might not be best for dynamic images

LFBOS (Gordon) -angular resolution
    -colour encoded with direction
    -qualitatively good
    -quantitatively not as robust as desired
-more robust than defocus
-have not done tomographic application
-have done some water surface reconstruction
-smooth glass with little diffraction
-larger diffraction better

Problems:
-might not be perfectly aligned
-camera gamut
-object causing diffusion

Background-oriented Schlieren
-for more information on Traditional Schlieren and BOS watch video for LFBOS paper
-optical inhomogeneities
-failing for large refraction indices
    -changes focus, get areas out of fucus



Traditional Schlieren (Fourier Optics)
-might want to look at work done for Disney
-used to map speech effects on air
-also Rainbow Schlieren

-knife edge is in Fourier space
-pathways that refract upwards through the object avoid the knife edge and make it to the camera's sensor array: more light when rays refracted upwards
-pathways that refract downwards before exiting the object are blocked by the knife edge: less light for rays refracted downwards

- **tomography on gradients** (linear system) **and Poission solver** (linear)
- both linear: $\int$ and ∇
- can these two be combined (since they are both linear)?
- **combined 2D/3D solver**
    - most likely not linear anymore
    - mapping colour -> directions
    - need to model noise
- also need to look at integration boundaries:

$$\int_{s^{min}}^{s^{max}}$$

- if we choose s^{min} and s^{max} values appropriately, we can reverse the order to: $\nabla\int$
    - can be used if these two points (integration boundaries) are on the same refractive index

- can take 2D displacement vectors
    - they are integrable
    - poisson solve -> 2D density (hull)
    - do tomography on 2D densities

- **integrate in 2D -> then tomography**
    - sparsity priority on 2D displacement vectors

- **all-in-one-system**
    - may be too big and impractical
    - keep track of uncertainties
    - order of operations and approximations might have an effect

- linear mapping of noise
- small changes in background map to small changes measurement noise (measurement offset)
    - depends on solver?
- local error -> small change
- expect to get accurate results for most pixels and then deal with outliers

- refractive index setup?
- camera setup?
    - new methods?

- 2D -> 3D problems with scalar division
    - limit refractive index

- take into consideration whole volume that can affect outcome of reconstruction volume
    - do not ignore errors caused in the surrounding environment
    - e.g. In the tank, between the camera and desired volume reconstruction area