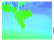
Novel Parameterizations in Structure and Motion

Guy Rosman [1]

July 8, 2013

[1] Joint work with Yu Wang, Anastasia Dubrovina, Alex and Michael Bronstein, Xue-Cheng Tai, Freddy Bruckstein and Ron Kimmel.

Topics

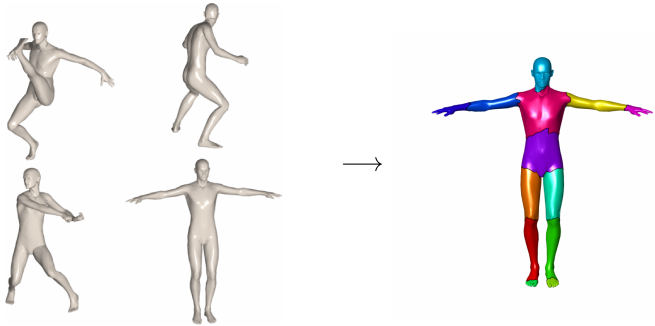
- In many cases of structure and motion understanding, choice of the right parameterization is critical.
- One way to go around this is to learn a good parametrization.
- In many cases, the right parameterization leads to elegant solutions.
- In some cases, allow fast computation.
- We demonstrate examples
  - Optical flow in stereo (SSVM '11)
  -  Articulated motion segmentation (SSVM '11, 3DOR '11, ECCV '12, NORDIA '12)
  - Structured light 3D reconstruction (3DIMPVT '12)

Fast Matrix-valued Regularization for Motion Segmentation

- The problem we started from: articulated motion segmentation and estimation on point clouds and depth video.
- We relate articulated motion segmentation and fast matrix- / group-valued image regularization.
- The new framework leads to motion segmentation, orientation denoising and diffusion tensor images (DTI) reconstruction.

Articulated Motion Segmentation

- We aim at segmenting articulated motion in $3D$ point clouds.
- We have 2 or more partial scans of the same $3D$ object, $\mathcal{S}_1, \mathcal{S}_2, \cdots$.
- We would like to obtain the rigid segments and their transformations with minimal additional assumptions.

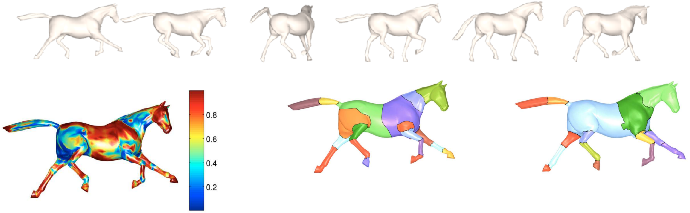


Articulated Motion Segmentation

- Many algorithms are available for articulated motion segmentation:
  - [Anguelov et al., UAI, '04] define EM steps over labeling and transformations,
  - [James et al., SIGGRAPH '05] perform mean-shift clustering,
  - [Huang et al., SIGGRAPH'08] perform clustering and deformation of the cluster points,
  - [Thierny et al., ICPR '08] measure changes in edge-lengths,
  - [Wuhrer and Brunton, VC '10] perform a segmentation of the dual graph based on the dihedral angles on neighboring triangles.
- Many algorithms incorporating cues other than motion in segmentation.

Articulated Motion Segmentation

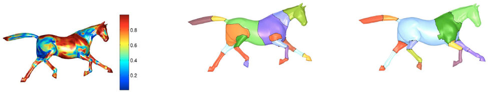
- We seek an implicit, bottom-up approach to segmentation,
- Unknown number of rigid parts, rigid parts are not always well-defined.
- Inside a rigid part - all points undergo the same rigid transformation.
- This leads to edge-preserving regularization of maps from the object' surface to the group of rigid motions ($SE(3)$)



Articulated Motion Segmentation

[R., Bronstein, Bronstein, Wolf, Kimmel, SSVM'11] - formulation as a Mumford-Shah/Ambrosio-Tortorelli segmentation of $SE(3)$ maps on the object, for triangulated surfaces

[R., Bronstein, Bronstein, Kimmel, 3DOR'12] - defined on point clouds, with depth images.

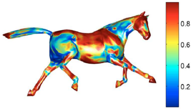


Interlude: Ambrosio-Tortorelli $SE(3)$ regularization

- An Ambrosio-Tortorelli (AT) scheme for piecewise-smooth $SE(3)$ regularization on a surface

$$\rho_{AT}(g) = \int_{\mathcal{S}} \left(\frac{1}{2} v^2 \|\nabla_X g\|^2 + \epsilon \|\nabla_X v\|^2 + \frac{(1-v)^2}{4\epsilon} \right) da$$

- g is a function $g : \mathcal{S} \to SE(3)$,
- v is a diffusivity term, close to 0 at region boundaries
- v is mostly equal to 1 and is spatially regular.



Fast Regularization of Matrix-Valued Images

- The AT scheme is slow to compute - requires explicit iterations using the Lie-algebra.
- We want an efficient solution.
- We have depth images as data in many cases - Cartesian grids are available.
- We re-formulate the problem as edge-preserving regularization for matrix-valued images.

Fast Regularization of Matrix-Valued Images

Matrix/group valued data is everywhere!

- Robotics and motion ([Park et al., '95], [Zefran and Kumar, '98], [Tuzel et al., '08], [Kobilarov et al., '09])
- Computer vision ([Perona, '98], [Tang and Sapiro '00], [Sochen and Kimmel '01], [Brox and Weickert '02])
- Medical images ([Stejskal and Tanner '65], [Basser et al., '94], [Tschumperle and Deriche '05], [Burgeth et al., '07], [Franken et al. '07], [Gur and Sochen '09], many more)

Functional Description

1. We start with a regularity term defined according to the Lie-algebra:

$$\min_{u \in \mathcal{G}} \int \|u^{-1}\nabla u\| + \lambda \|u - u_0\|^2 dx,$$

2. If $u \in \mathcal{G}$, for Lie-groups such as $SO(n), SE(n)$, we can simplify:

$$\min_{u \in \mathcal{G}} \int \|\nabla u\| + \lambda \|u - u_0\|^2 dx,$$

where ∇u is the Jacobian of the mapping from the image into an embedding space, for example $\mathbb{R}^9$ for $SO(3)$ (Also possible for $SPD(n)$ images, and other matrix manifolds)
- Another option – second-order regularization

$$\min_{u \in \mathcal{G}} \int \|Hu\| + \lambda \|u - u_0\|^2 dx,$$

Algorithm Description

3. We now add an auxiliary variable $v \in \mathcal{G}$, and constrain $u = v$ (but remove the restriction on $u \in \mathcal{G}$).
4. We constrain v to equal u using an augmented Lagrangian term,

$$\min_{v \in \mathcal{G},\quad u} \int \|\nabla u\| + \lambda \|u - u_0\|^2 + \langle \mu, u - v \rangle + \frac{r}{2} (u - v)^2 dx,$$

5. Total variation (TV) denoising w.r.t. u - (Wen/Goldfarb/Yin '09, Tai '09, other possibilities as well)
(We can use other quadratic fitting/reconstruction terms) (Also, other regularization terms)

Algorithm Description

6. Optimizing w.r.t. v under constraint is a projection step per pixel,

$$\underset{v \in \mathcal{G}}{\operatorname{argmin}} \langle \mu, u - v \rangle + \frac{r}{2} (u - v)^2 = \underset{\mathcal{G}}{\operatorname{Proj}} \left(u + \frac{\mu}{r} \right) \tag{1}$$

Algorithm Description

The overall algorithm is as follows
Repeat for $k = 0, 1, \ldots$ until convergence

1. update $F(u, v, \mu)$ w.r.t u - fast TV regularization step
2. update $F(u, v, \mu)$ w.r.t v - using eq. 1 (projection step)
3. update The lagrange multipliers using

$$\mu^k = \mu^{k-1} + r\left(v^k - u^k\right)$$

Convergence Properties

We look at the case where the Lagrange multipliers are fixed,

- The projection operator is discontinuous

$$\left(\underset{SO(n)}{\mathrm{Proj}}\,(\mathbf{0})\right)$$

- This makes a straightforward convergence proof difficult
- Instead, we slightly modify the algorithm (as suggested by Attouch et al. , '10)

$$\left(u^{k+1}, v^{k+1}\right) = \underset{v \in \mathcal{G}, u}{\operatorname{argmin}} \int \begin{array}{c} \|\nabla u\| + \lambda \|u - u_0\|^2 + \langle \mu, u - v \rangle + \frac{r}{2}(u - v)^2 \\ + \frac{1}{\theta}\|u - u^k\|^2 + \frac{1}{\theta}\|v - v^k\|^2 \end{array} dx,$$

Convergence Properties

- Looking at the two optimization problems,

$$
\begin{aligned}
u^k &= \operatorname*{argmin}_{u} F(u, v^{k-1}, \mu) + \frac{1}{\theta_k}\|u - u^{k-1}\|^2 \\
v^k &= \operatorname*{argmin}_{v \in \mathcal{G}} F(u^k, v, \mu) + \frac{1}{\theta_k}\|v - v^{k-1}\|^2.
\end{aligned}
$$

- And since $F(u, v, \mu) > -\infty$, We can show convergence, since

$$
\sum_k^{\infty} \frac{1}{\theta_k}(\|u^k - u^{k-1}\|^2 + \|v^k - v^{k-1}\|^2) < \infty.
$$

- And $L(u, v)$ is bounded from below

Convergence Properties

- Even though the domain is non-convex, we can show convergence with small modifications.
- For $SPD(n)$, since the domain is convex, the whole problem becomes a convex optimization problem.
- This easily shows convergence given μ
- There is also convergence proof for the complete AL method in this case.

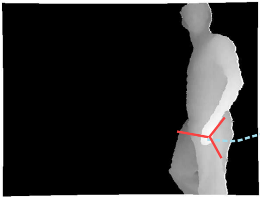
Results

Turns out, this formulation is important in several applications:

1. $SE(3)$ - Segmentation and motion scale-space from depth sensors
2. $SO(2)$ - Direction diffusion ($SO(2)$ is isomorphic to S^1)
3. $SPD(3)$ - DTI regularization and regularized reconstruction

Motion Scale-space

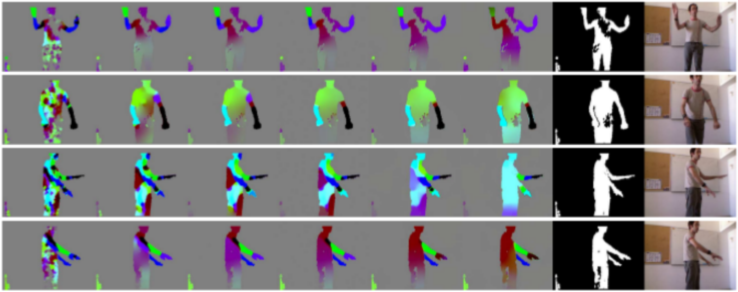
1. 3D frames are taken with a depth sensors.



2. Local rigid motion is computed at each point
3. $SE(3)$ diffusion scale-space is used to discern major moving parts.
4. Fast regularization: for a 320×240 image, convergence takes $49ms$ (GTX 580).

Motion Scale-space

Simple (local) motion estimation - leads to noisy results:

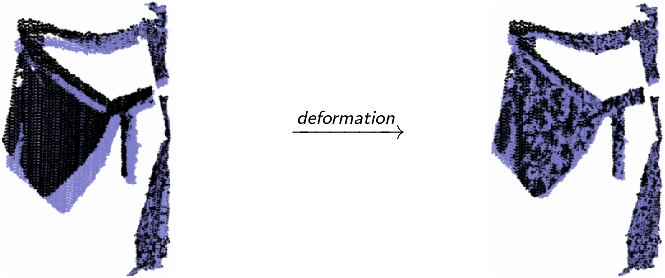


(Different colors visualize different rigid motions)

1. Fast regularization: for a 320×240 image, convergence takes $49ms$ (GTX 580).

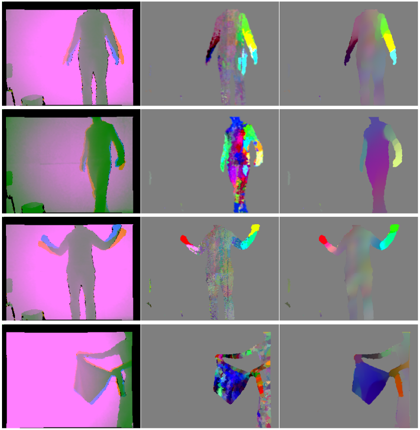
Motion Scale-space

Using better motion estimation, based on nonrigid-deformation:



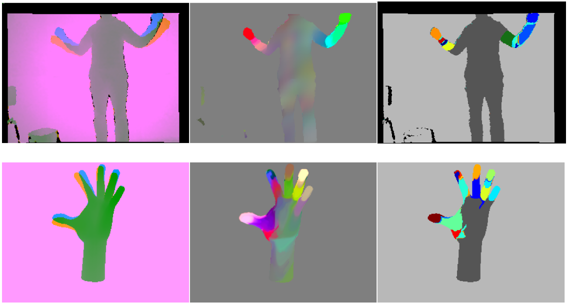
Motion Scale-space

Use local least-squares fitting to get the initial rotation and translation:



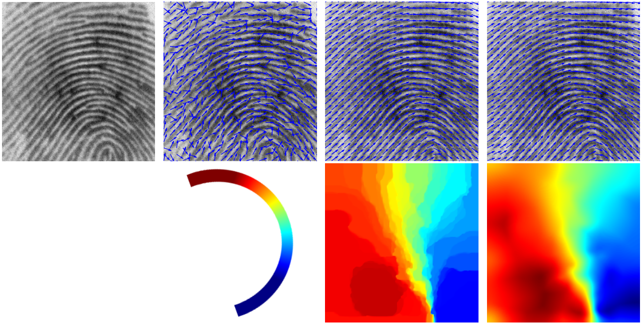
Motion Scale-space

Can we really get segmentation out of it? Yes we can!
Allows to obtain articulated parts (for example, using mean-shift clustering)



Orientation Diffusion

1. Another example - fingerprint curve smoothing ($SO(2)$ / S^1)



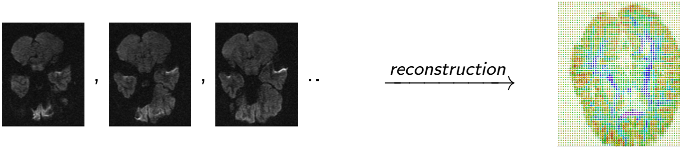
MSE for noised fingerprint image was
0.0270 for the second-order regularization,
0.0317 for the 1st-order / TV functional, (staircasing artifacts..)
0.0324 by [SK'01],
0.0449 in the original image.

Diffusion Tensor Imaging

1. Another application - diffusion-tensor image reconstruction (and denoising)
2. The convex domain allows us to show global convergence
3. Basically our data term comes from the Stejskal-Tanner equation

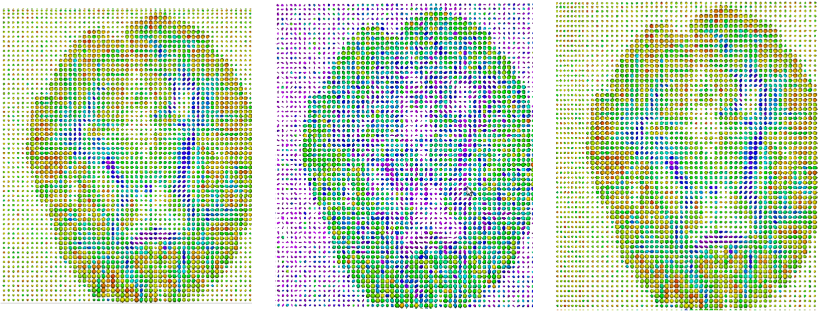
$$\operatorname*{argmin}_{v \in SPD(n)} \int \begin{array}{c} \sum_i \left\| b_i g_i^T u g_i - \log\left(\frac{S_i}{S_0}\right) \right\|^2 + \\ \lambda \|\nabla u\| + \frac{r}{2}\|v - u\|^2 + \langle \mu, v - u \rangle dx. \end{array}$$

(g_i - measurement vectors, S_i - DWI measurement from MR scan i)



Diffusion Tensor Imaging

Here we show regularized reconstruction from a noisy (AGWN) set of diffusion-weighted images (linear reconstruction data term)



Recap - part I

1. We formulate bottom-up articulated motion segmentation as matrix-valued TV regularization.
2. Using the augmented-Lagrangian technique we obtain a simple and efficient algorithm for regularization of matrix-valued images in general.
3. Achieves real-time performance for bottom-up articulated motion segmentation with minimal assumptions.
4. Examples show its usefulness for several applications.

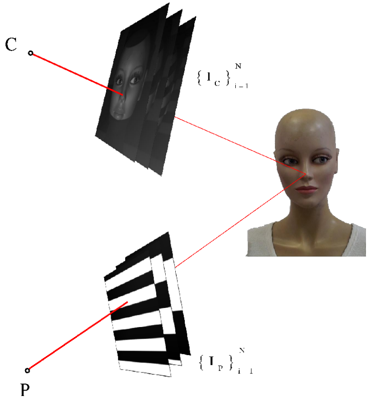
Sparse Priors for Structured Light

1. A lot of attention has been given to 3D reconstruction and 3D scanners
2. In close-range applications, structured-light is a highly important technique.

3. Time-multiplexed structured light is limited by camera exposure time and projector power limits.

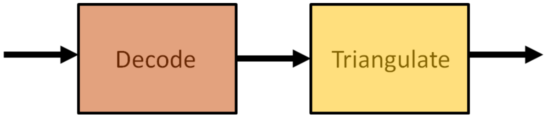
Sparse Priors for Structured Light

1. A structured-light setup includes a (calibrated) projector P and camera C pair

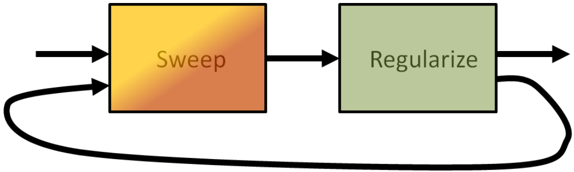


Sparse Priors for Structured Light

- Instead of the usual structured light reconstruction pipeline:



- We suggest



Structured Light - Probabilistic Approach

1. Given the correct depth (hence - camera/projector coordinates), the transformation between the projector and camera intensity is (assuming no gain / gamma correction, and relatively low noise)

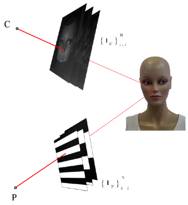
$$I_C^{(i)}(\mathbf{x}) = a(\mathbf{x}) I_P^{(i)}\left(\Pi_z(\mathbf{x})\right) + b(\mathbf{x}) + n^{(i)}\left(\mathbf{x}\right),$$

- a, b - dependent on surface normal and lighting direction, ambient light
- n - approximated by Gaussian noise in time-multiplexed SL.
- $\Pi_z(\mathbf{x})$ denotes the backprojection and projection based on depth z.

Structured Light - Probabilistic Approach

1. This allows us to rewrite structured light as a correlation plane-sweeping (easy to parallelize to real-time)

$$\underset{z,a,b}{\operatorname{argmin}} \left(-logP\left(I_P, I_C, a, b|z\right)\right)$$



Structured Light - Probabilistic Approach

1. Now we can also incorporate a prior for the surface smoothness

$$\underset{z,a,b}{\operatorname{argmin}} \left(-logP\left(I_P, I_C, a, b|z\right) - logP\left(z\right)\right)$$

2. The prior is easily inserted as an auxiliary variable

$$\underset{z,\tilde{z},a,b}{\operatorname{argmin}} \int \left(\underbrace{-logP\left(I_P, I_C, a, b|z\right)}_{correlation} - \underbrace{logP\left(\tilde{z}\right)}_{smoothness} + \frac{r}{2}\left(z - \tilde{z}\right)^2 \right) dxdy$$

1. Each part of the optimization is simple to perform.

Structured Light - Total Variation Prior

1. Example – total variation,

$$\underset{z,\tilde{z},a,b}{\operatorname{argmin}} \int \left(-logP\left(I_P, I_C, a, b|z\right)\right) + \|\nabla \tilde{z}\| + \frac{r}{2}\left(z-\tilde{z}\right)^2 dxdy$$

1. Optimization w.r.t z / $\tilde{z}$ is global / convex (as in [Steinbruecker et al., '09]).

Structured Light - Sparsity-based Prior

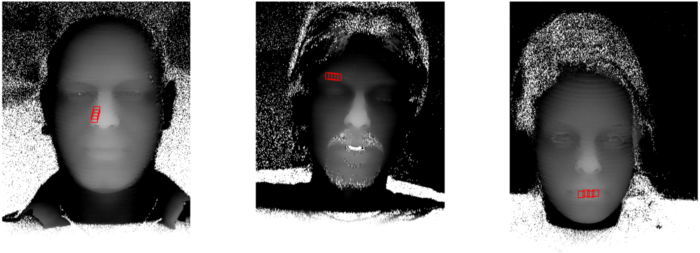
1. More powerful priors are usually patch based - for example L_1 sparsity

$$\underset{z,\alpha_i,a,b}{\operatorname{argmin}} \int \left(-logP\left(I_P, I_C, a, b|z\right)\right) + \frac{r}{2}\sum_i \left(P_i z - D\alpha_i\right)^2 + \|\alpha_i\| dxdy$$

1. Small patches are described as a sparse combination of dictionary patches.
2. Optimization w.r.t z / α_i is global / convex.

Structured Light - Sparsity-based Prior

- Motivation (1) - if we look at a range image, certain features repeat

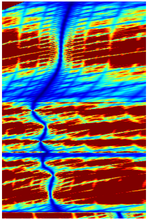


Structured Light - Sparsity-based Prior

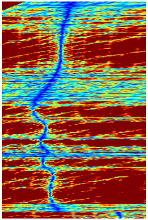
- Motivation (2) - sparsity - locally, range patches can be described as a sparse linear combination of dictionary atoms, describing the depth discontinuities and slopes.
- We take it into account in the sensing, rather than the preprocessing step (image denoising) or postprocessing step (range image denoising).
- Local priors alone are too weak compared to patch-based.

Structured Light - Sparsity-based Prior

- Motivation (3) - looking at the correlation cost as a function of depth and location, we can clearly see the “correct” surface



Low noise

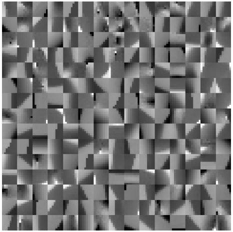


High noise

- Even though the best per-pixel solution may be incorrect..

Creating a dictionary from range-images

- In order to create a dictionary, we can use the KSVD algorithm [Aharon & Elad '06] on range images, but:
  - We must treat outliers.
  - We must address the rareness of "interesting" patches.
- The resulting dictionary captures mostly edges and slopes.



Algorithm Description

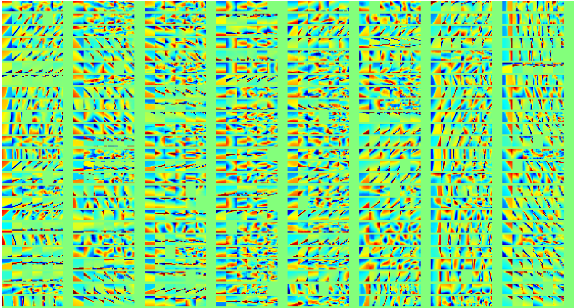
$$\underset{z,\alpha_i,a,b}{\operatorname{argmin}} \int \left(-logP\left(I_P, I_C, a, b|z\right)\right) + \frac{r}{2}\sum_i \left(P_i z - D\alpha_i\right)^2 + \|\alpha_i\| dxdy$$

- We solve for the depth z and the auxiliary variables α_i as follows

1. Initialize z by depth sweep
2. Given local patch $P_i z$, update α_i by iterative shrinkage
3. Given $\{\alpha_i\}_{i=1}^N$, update z by plane sweeping (with an added quadratic term per pixel)
4. Go back to step 2.

Gaussian-mixture model for range-images

- We can also use the GMM framework used in [Yu/Sapiro/Mallat, '9], [Zoran and Weiss, '11]
- Each 6-element row represents the principle axes of a GMM component
- We used 200 components/Gaussians – most of them represent edges, some corners.



Results

Two main problems in time-multiplexed structured light:

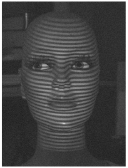
- High frame rate $\rightarrow$ image noise and decoding errors
- Low frame rate $\rightarrow$ motion artifacts

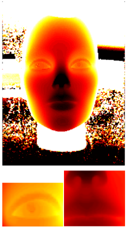
Two important questions:

- Can we use weaker priors?
- Can we just use sparsity for cleaning the noise? (Tosic/Olshausen/Culpepper, '10)

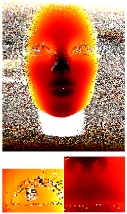
Noise Level	Raw L_2 error	Median L_2 error	TV Reconst. L_2 error	Sparse Denoising L_2 error	Sparse Reconst. L_2 error	Raw L_1 error	Median L_1 error	TV Reconst. L_1 error	Sparse Denoising L_1 error	Sparse Reconst. L_1 error
2.5	1.4608	0.8411	0.8744	0.8680	0.8191	0.5996	0.4255	0.4240	0.4298	0.3379
5	2.6443	1.1033	1.1508	1.1768	0.9584	1.2013	0.5696	0.5689	0.6356	0.4135
7.5	3.9080	1.5315	1.715	1.8136	1.3489	2.1032	0.7384	0.7164	0.9489	0.5603
10	4.9841	1.9399	2.3866	2.758	1.7490	3.0949	0.9840	1.216	1.288	0.7571

Results - Low Exposure

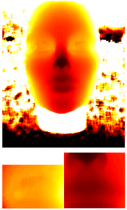




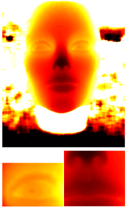
Ground truth



$\sigma = 5$



Median



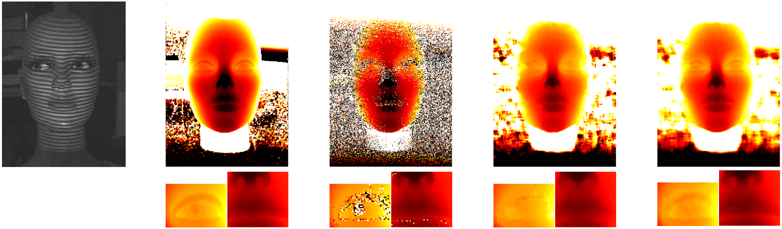
Sparse Prior

Raw

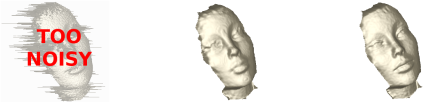
Median

Sparse Prior

Results - Low(er) Exposure

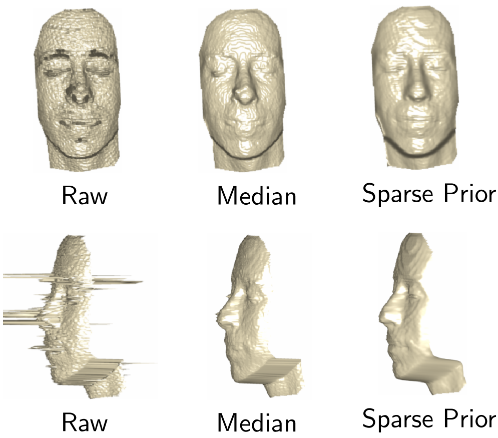


Ground truth | $\sigma = 10$ | Median | Sparse Prior



Raw | Median | Sparse Prior

Results - Motion Artifacts

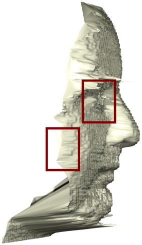


Results - Motion Artifacts

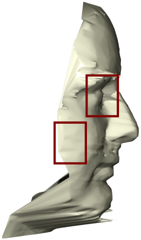
- Motion artifacts due to motion of sharp albedo changes around eyebrows, nose and mouth.
- Artifacts far from i.i.d noise, images deviate from our model.
- We are still able to correct the artifacts where median filtering fails.

Results - Real-life Exposure Artifacts

Camera Input



Median Postprocessed



2nd order TV Reconst.

Recap - part II

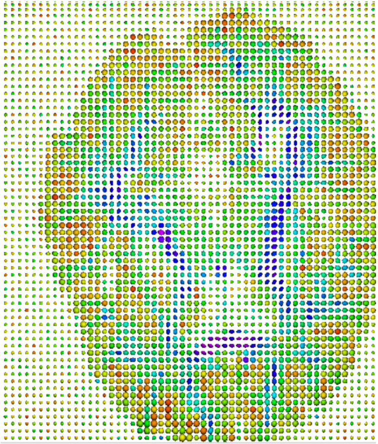
- Formulate structured light as plane-sweeping.
- Take a strong prior, alternating minimization helps.
- Insert the prior into the reconstruction.
- The resulting prior is quite intuitive.

Thank you for your attention!

Backup

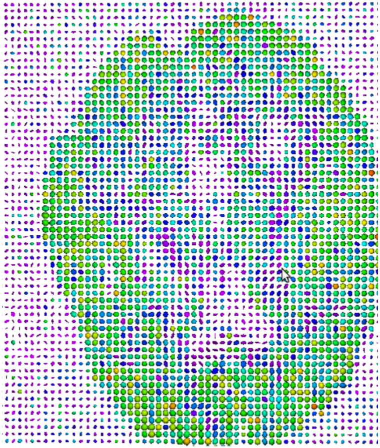
Diffusion Tensor Imaging

close-up: Original image



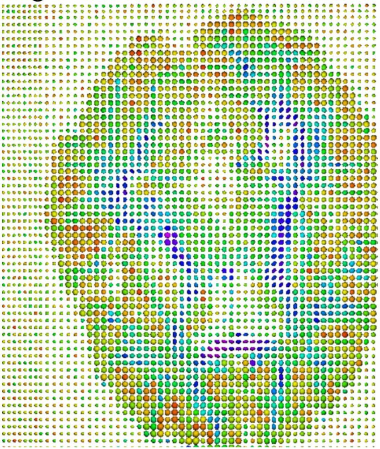
Diffusion Tensor Imaging

Reconstruction from noisy DWI



Diffusion Tensor Imaging

Regularized reconstruction



Motion Scale-space

The Stejskal-Tanner equation

$$\frac{S}{S_0} = \exp\left[-\gamma^2 G^2 \delta^2 \left(\Delta - \frac{\delta}{3}\right) D\right]$$

where

- γ - the Gyromagnetic constant, G - gradient strength
- δ - pulse length
- Δ - inter-pulse interval
- D - the diffusion coefficient, according to $g^T u g$