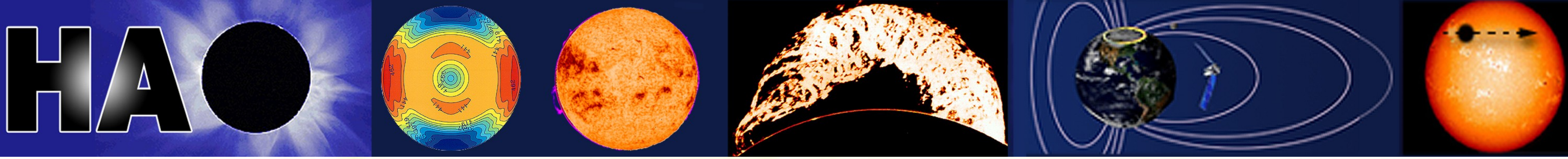


Stokes Inversion via Principal Component Analysis

R. Casini

HAO-NCAR

NCAR

1 February 2019

NLTE on the Sun

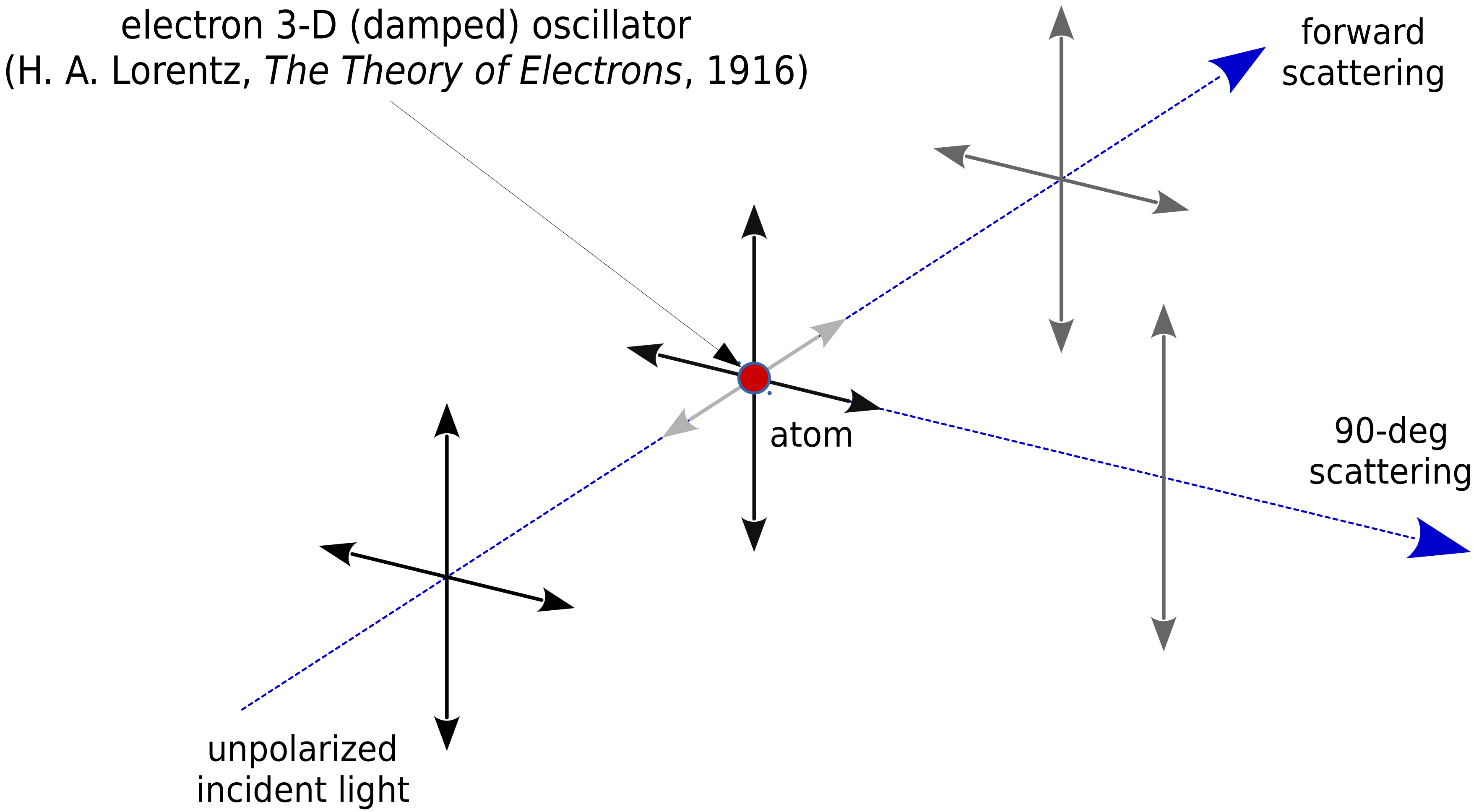
- temperature gradients in the stratified atmosphere cause anisotropic irradiation of the atmospheric layers

→ atomic polarization (i.e., *population imbalance* and *quantum coherence* among magnetic sub-levels)

- density gradients drive the thermalization of the atomic populations

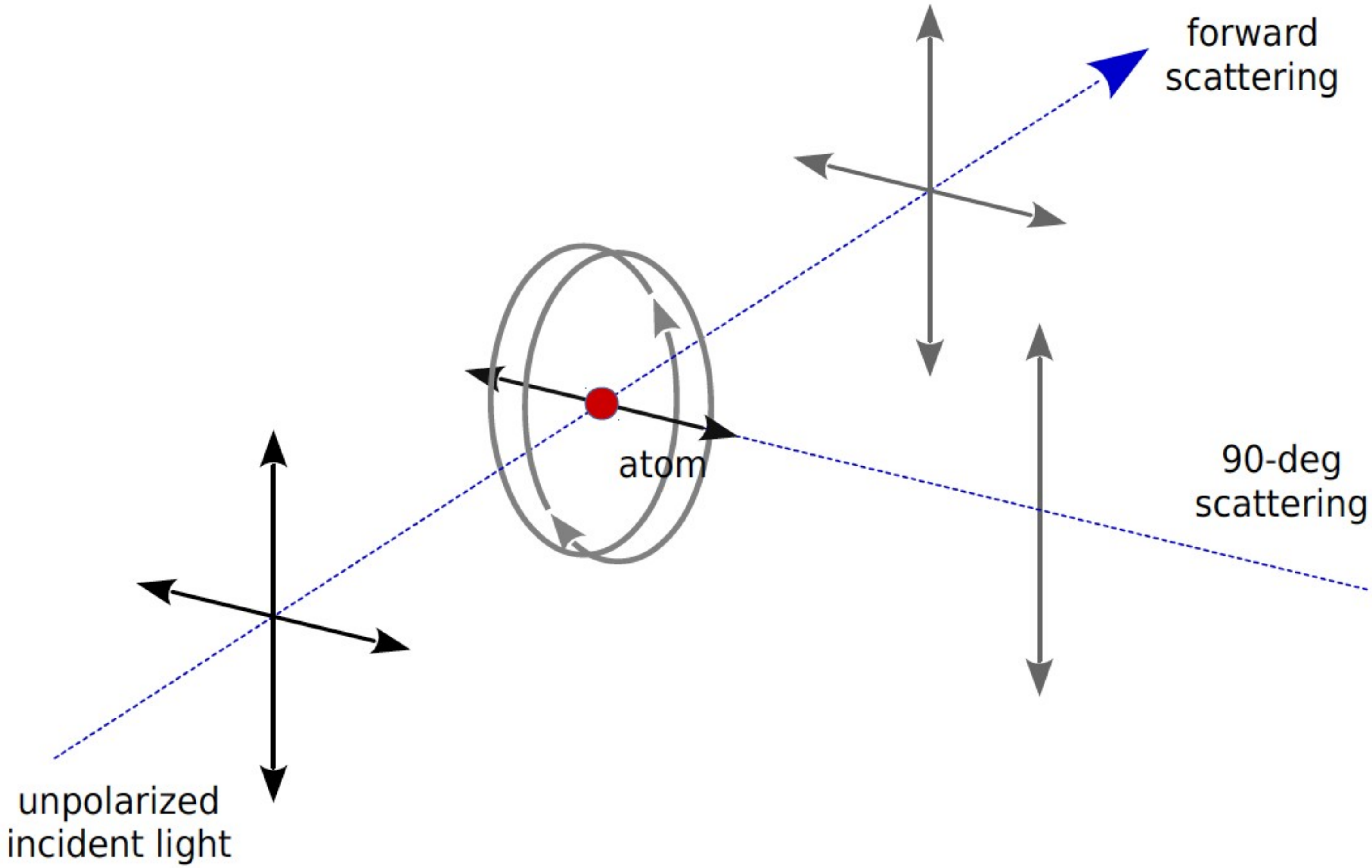
→ competing effects of anisotropic irradiation and isotropic collisions

Scattering Polarization
(semi-classical view)



e.g., Rayleigh scattering

Scattering Polarization
(semi-classical view)



equivalent description in spherical basis

NLTE on the Sun

- photosphere: radiation anisotropy is small; collision isotropy is high; collisional rates are large

  → atomic polarization is typically negligible (with some notable exceptions)

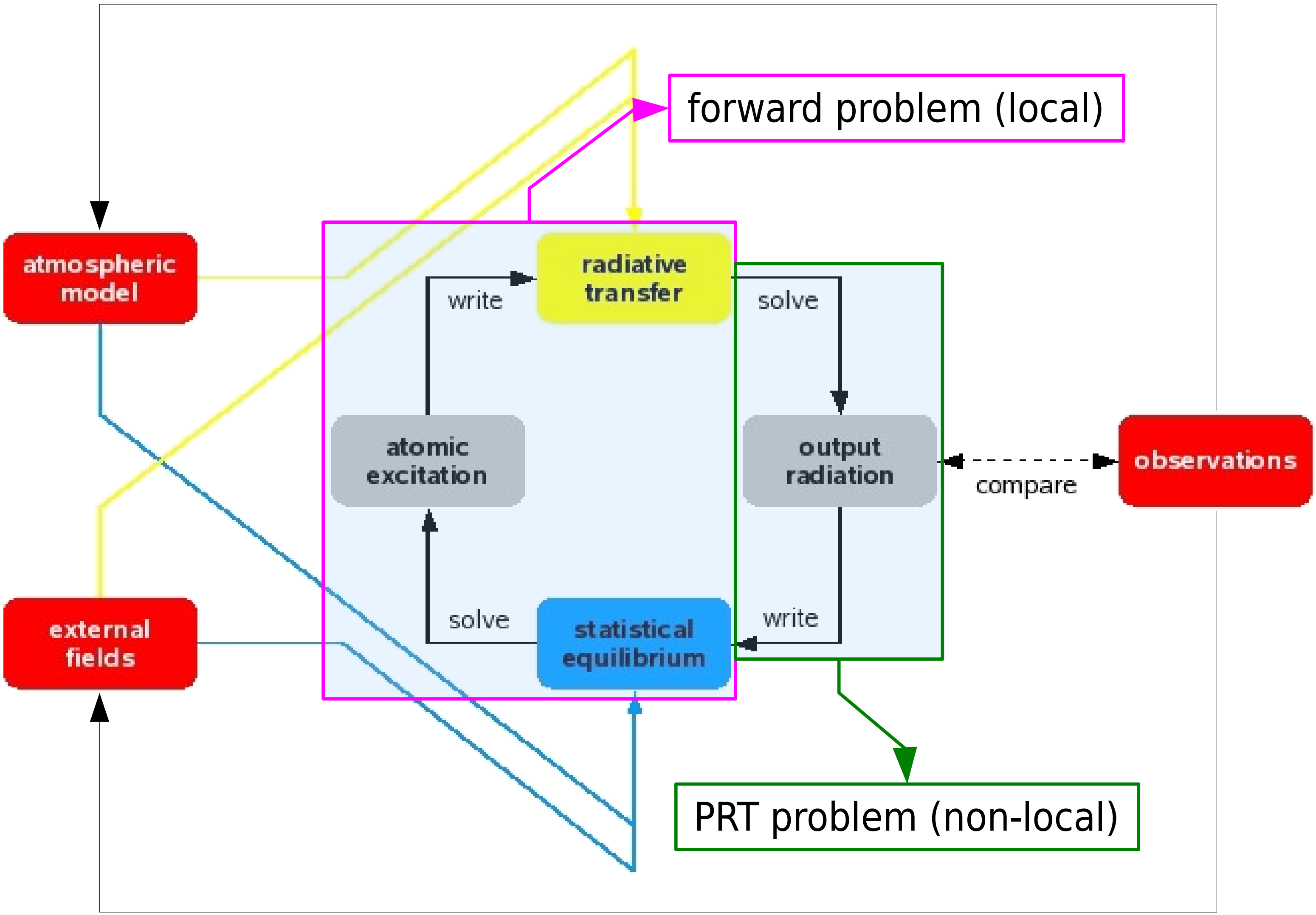
- chromosphere: radiation anisotropy may be large (mostly dependent on CLV); collision isotropy is high; collisional rates decrease quickly with height

  → atomic polarization is important

- corona: radiation anisotropy is dominant (from both CLV and height); collision isotropy starts to break down; collisional rates are low

  → atomic polarization is dominant

NLTE on the Sun



“self-consistency” loop

(Landi Degl’Innocenti & Landolfi 2004)

Stokes Inversion via Pattern Recognition

the NLTE "inverse" problem is built upon a complex and time-consuming forward problem

→ inversion strategy: pattern recognition techniques
Principal Component Analysis (PCA)

Stokes Inversion via Pattern Recognition

the NLTE "inverse" problem is built upon a complex and time-consuming forward problem

→ inversion strategy: pattern recognition techniques Principal Component Analysis (PCA)

Procedure

- determine a "universal" basis of Stokes eigenprofiles (by some optimized sampling of parameter space)

→ projection coefficients define a "dual" Stokes space

- build a database of Stokes profiles that is "uniformely" dense in this dual space (by "filtered" Monte Carlo)
- project observations over the eigenbasis
- match observations to database entries (e.g., minimize Euclidean distance in the dual space)

Principal Component Analysis
Single-Line

Example

He I λ1083 nm

on-disk
($20° \leqslant \vartheta \leqslant 30°$)

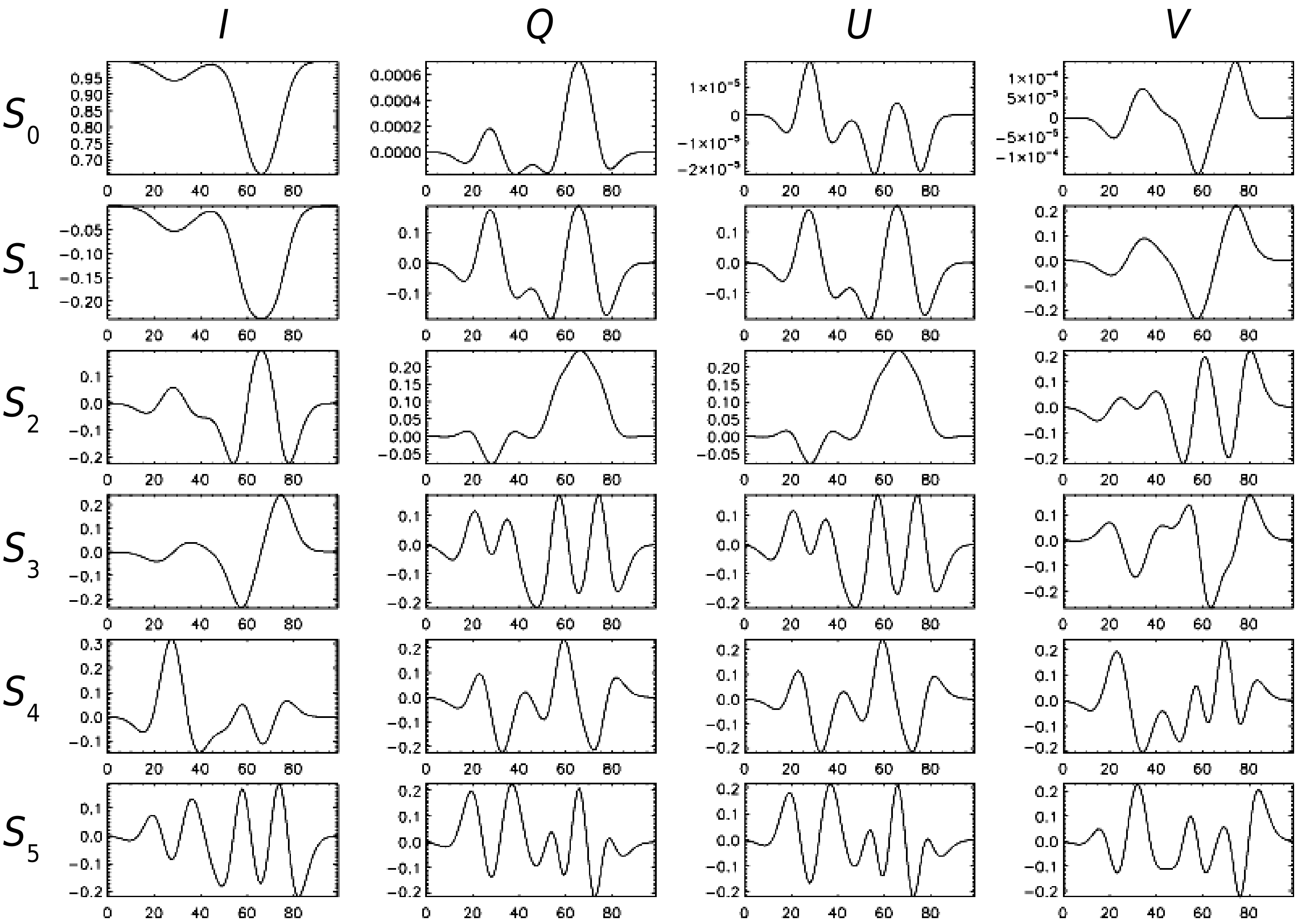
$h \leqslant 0.06\ R_\odot$

$B \leqslant 2000$ G

(ϑ_B, φ_B) in 4π srad

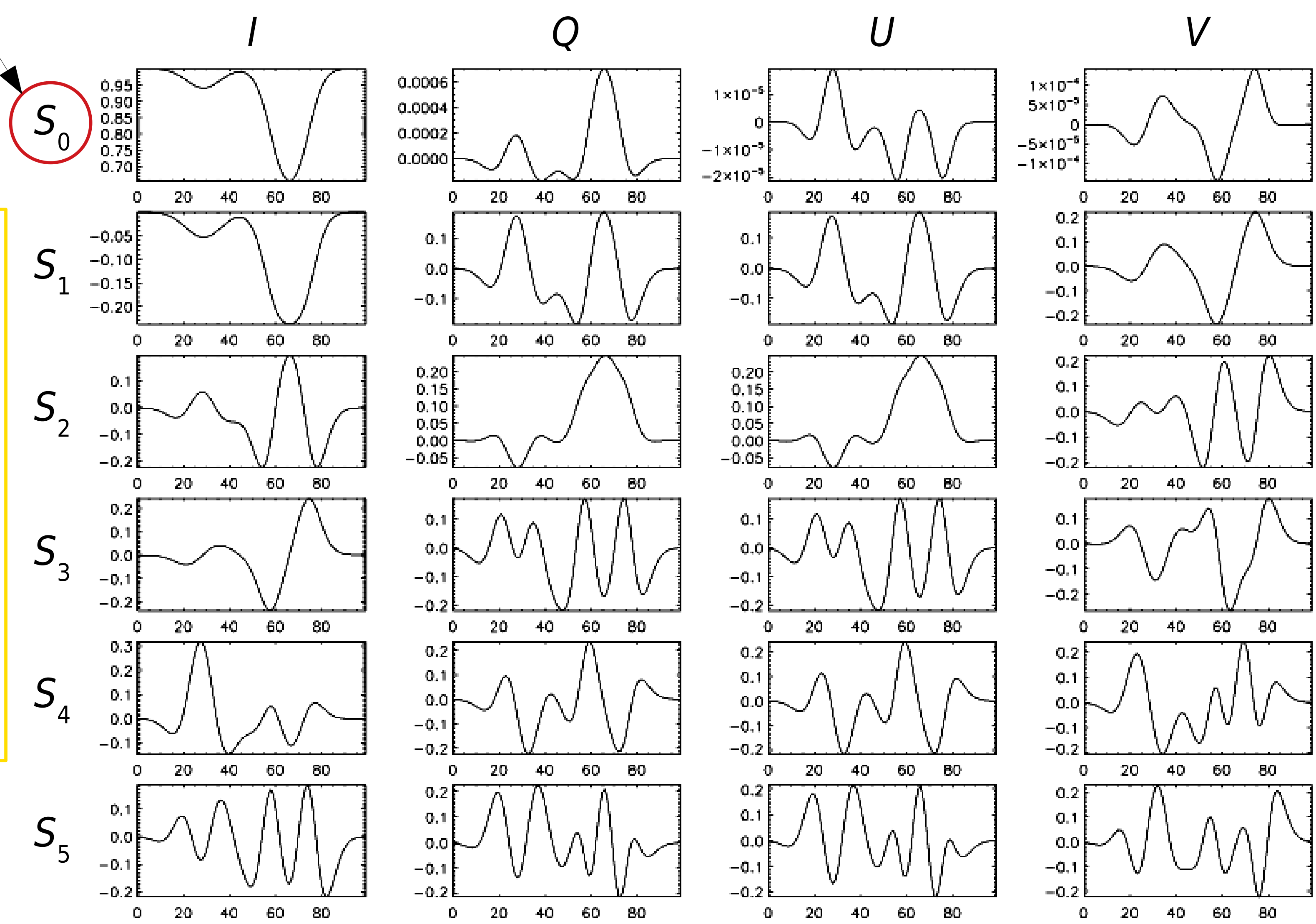
10^4 K $\leqslant T \leqslant 2\times10^4$ K

$0.2 \leqslant \tau \leqslant 1.4$



Principal Component Analysis
Single-Line

Average over parameter space



Example

He I λ1083 nm

on-disk
(20°⩽ϑ⩽30°)

$h \leqslant 0.06\ R_\odot$

$B \leqslant 2000$ G

(ϑ_B,φ_B) in 4π srad

10^4 K$\leqslant T \leqslant 2\times10^4$ K

$0.2\leqslant\tau\leqslant1.4$

Principal Component Analysis

Multi-Line

Example

He I λ1083 nm

+ λ587.6 nm

off-limb
($85° \leqslant \vartheta \leqslant 95°$)

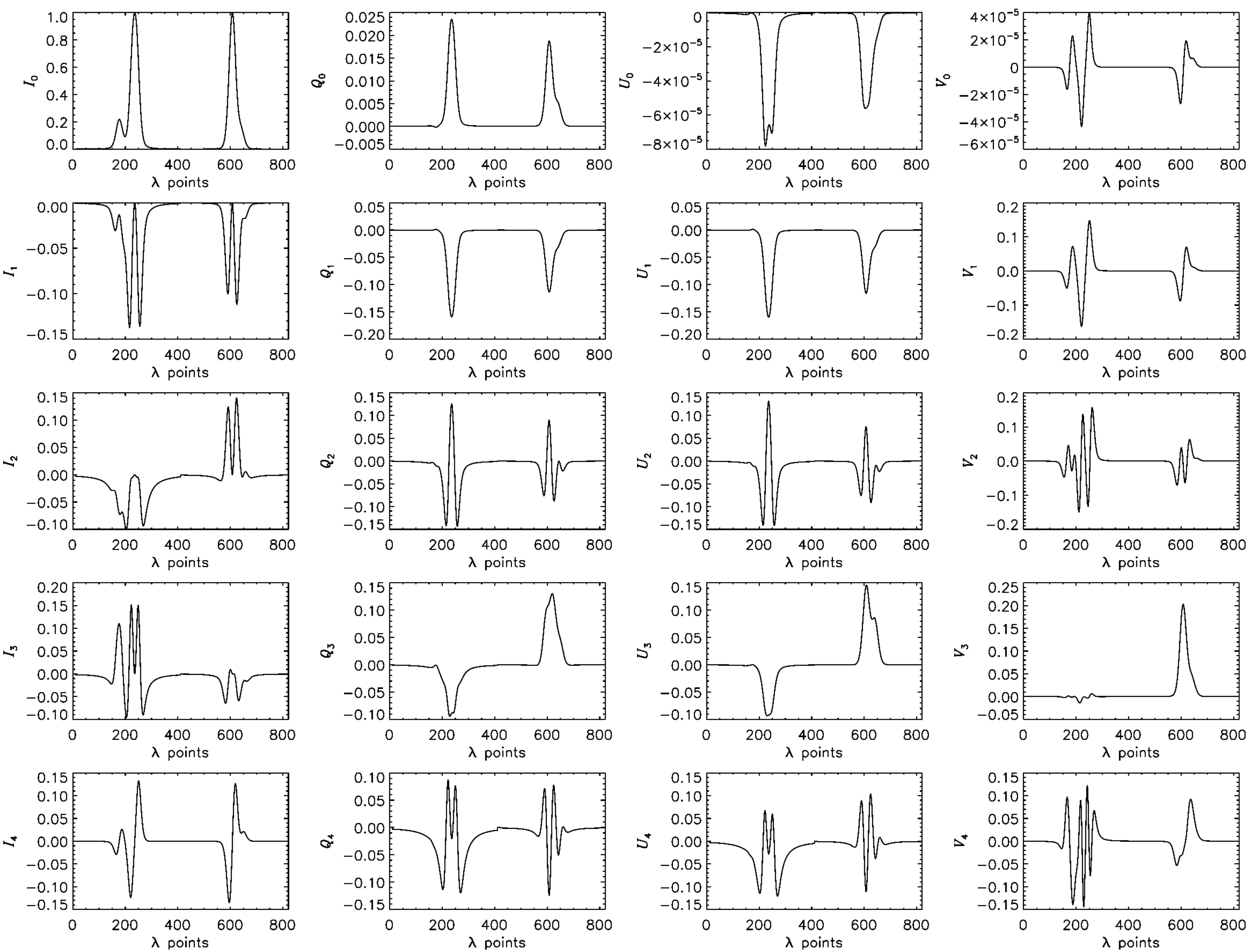
$0.01\ R_\odot \leqslant h \leqslant 0.06\ R_\odot$

$B \leqslant 200$ G

(ϑ_B, φ_B) in 4π srad

10^4 K $\leqslant T \leqslant 1.5\times10^4$ K

$0.5 \leqslant \tau \leqslant 1.5$ (λ1083)



PCA: General Considerations

Pros

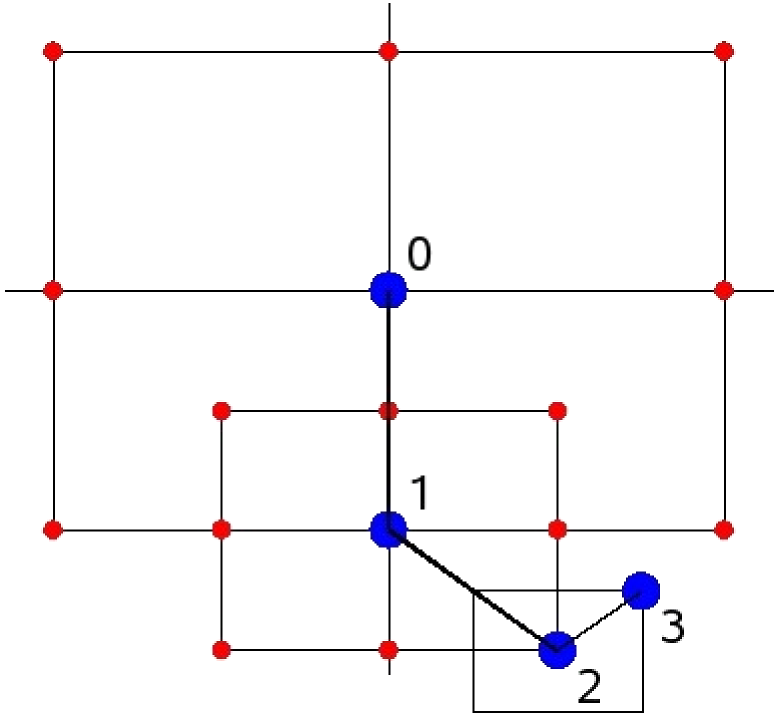
- fast (searches best fit in a pre-built database of models)
- stable (always finds best fit: no issues with local minima)
- model independent (universal search/minimization algorithm)

Cons

- no solution refinement (errors fixed by the density of the database)
- database can become unmanageably large (dimensionality of parameter space, parameter ranges, target error; partial mitigation from optimally sampling the parameter space, indexing)

PCA: Solution Refinement

simple approach to solution refinement



- "0"-solution and initial error box are determined by the PCA inversion
- *N*-step refinement by search of the error perimeter (e.g., halving the length at each step)

Issues

- possible trapping in local minima (if initial error box is too large)

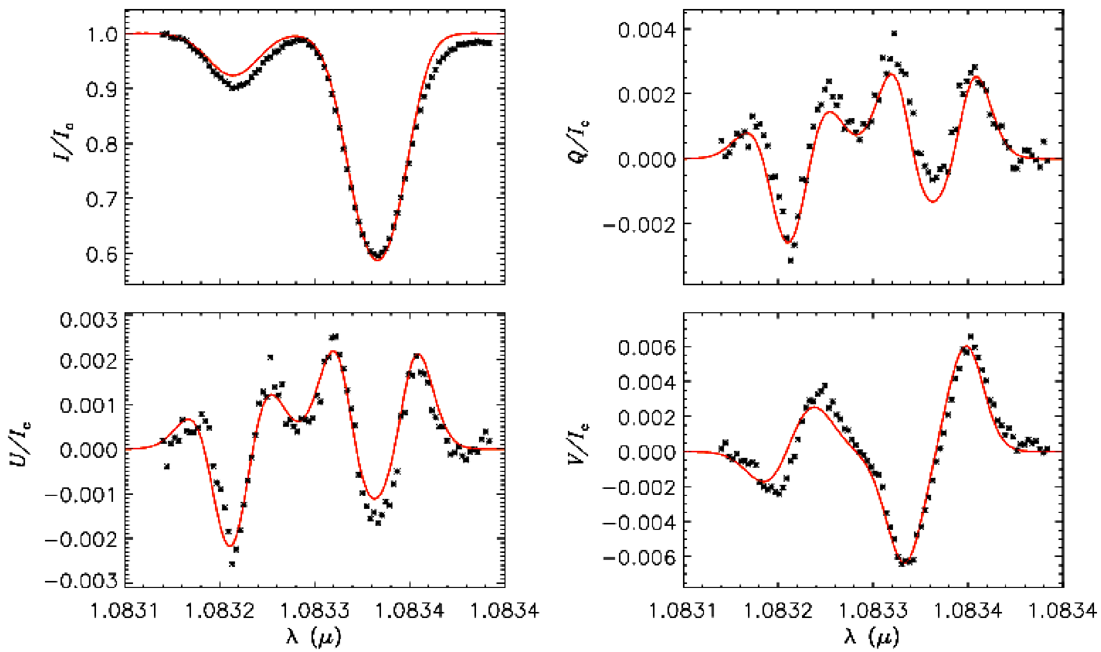
→ need PCA database sufficiently dense for initial inversion

- systematic search of error perimeter can be very slow (depends on number of parameters in forward model)

→ need better strategy (e.g., Levenberg-Marquardt minimization)

Magnetic map of an A-R filament

He I λ1083 nm
Martínez Pillet, July 5, 2005
@ VTT + IAC TIP II



τ = 0.958 ± 0.035

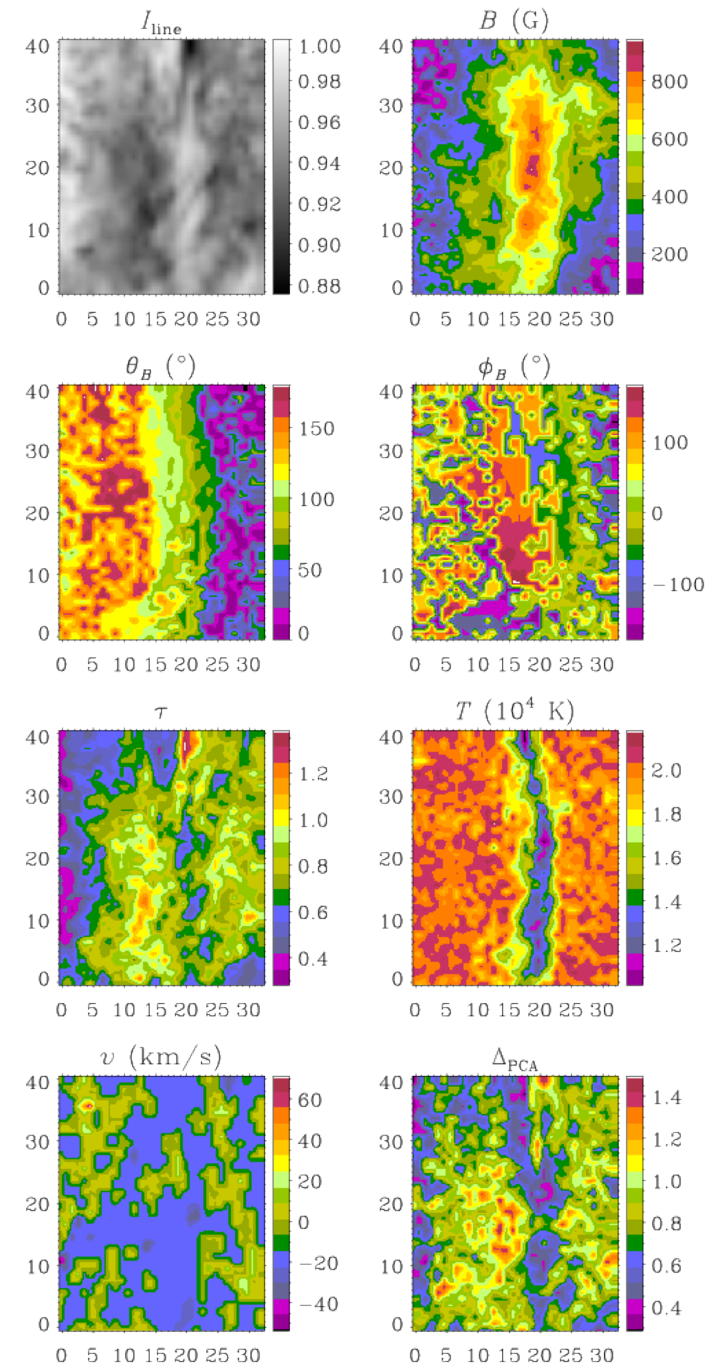
T = 13534.70 ± 521.18 K
δ = 0.736

θ = 27.04 ± 0.76
h = 0.0338 ± 0.0038 R

B = 733.05 ± 22.96 G

θ_B = 104.89 ± 0.99
ϕ_B = −74.92 ± 0.99

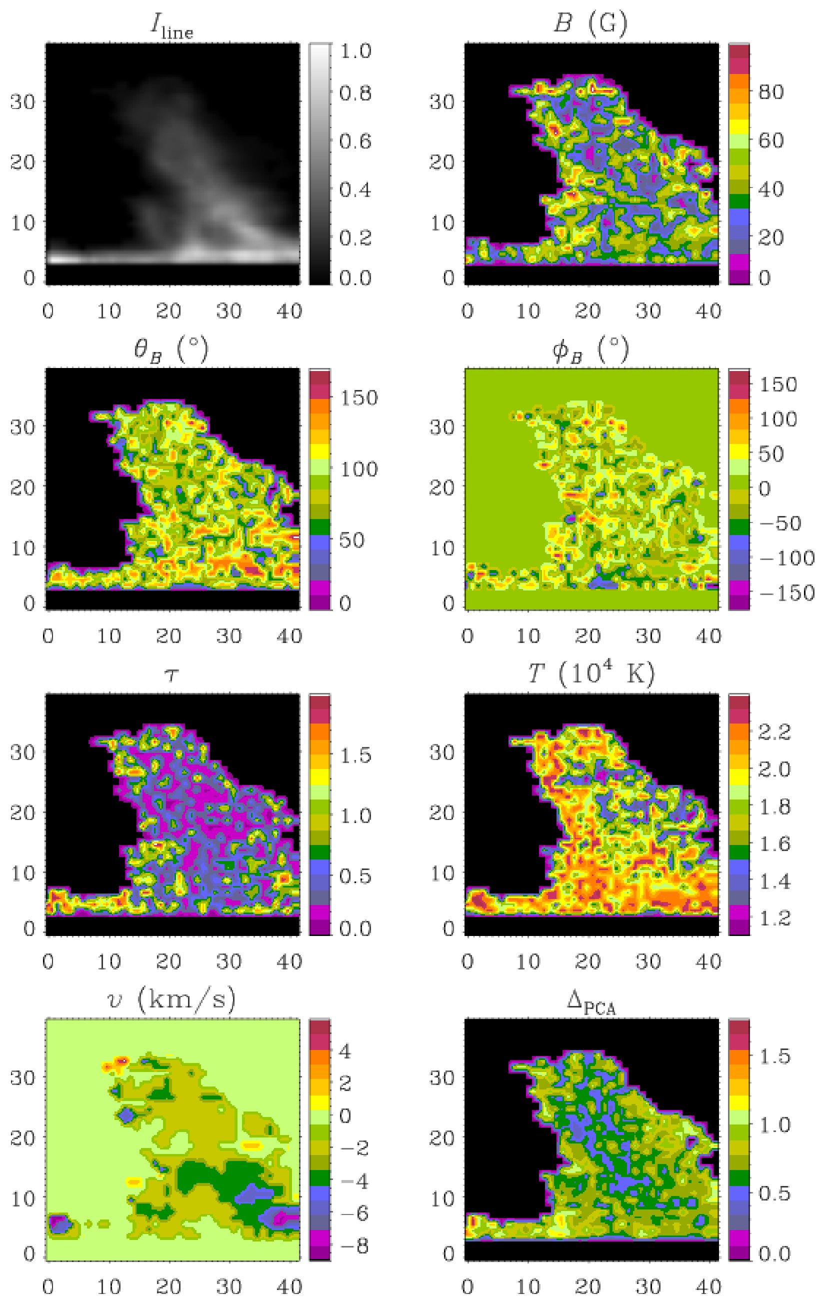
Θ_B = 96.58 ± 0.49
Φ_B = −69.93 ± 23.87



Kuckein et al. 2009

Magnetic map of a quiescent prominence

He I λ587.6 nm (D_3)
Casini et al., May 25, 2002
@ DST + HAO ASP



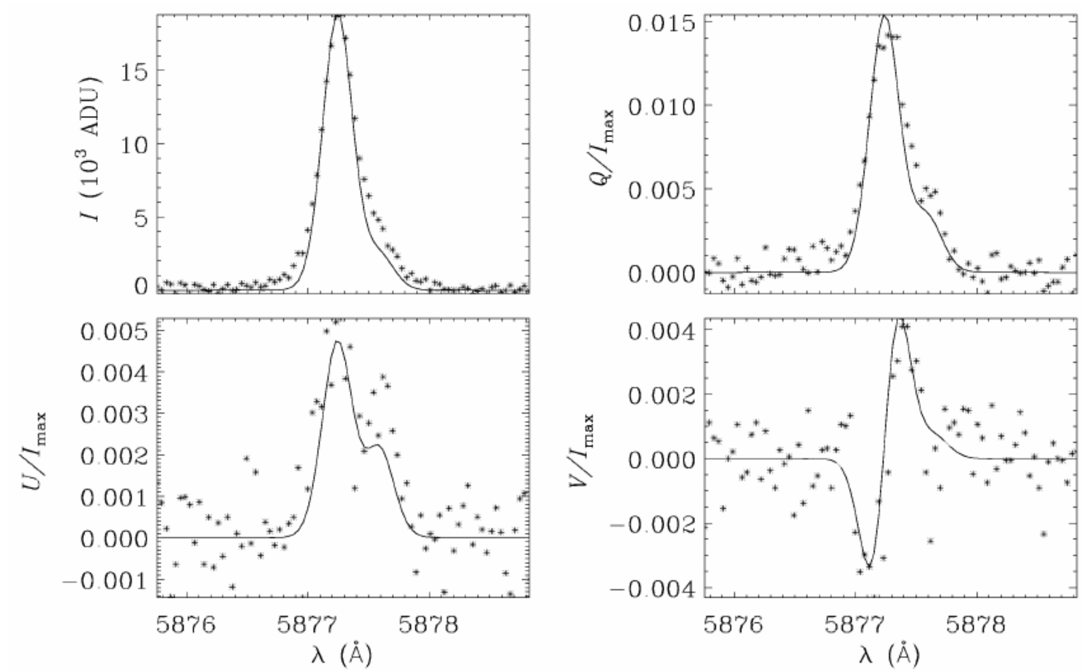


FIG. 6.—Same as Fig. 2, but for the point (15, 26) in the maps, corresponding to a field strength of 87 G ($\Theta_B = 56°$). Because of the larger field strength, the antisymmetric shape of Stokes V is now evident.

Casini et al. 2003

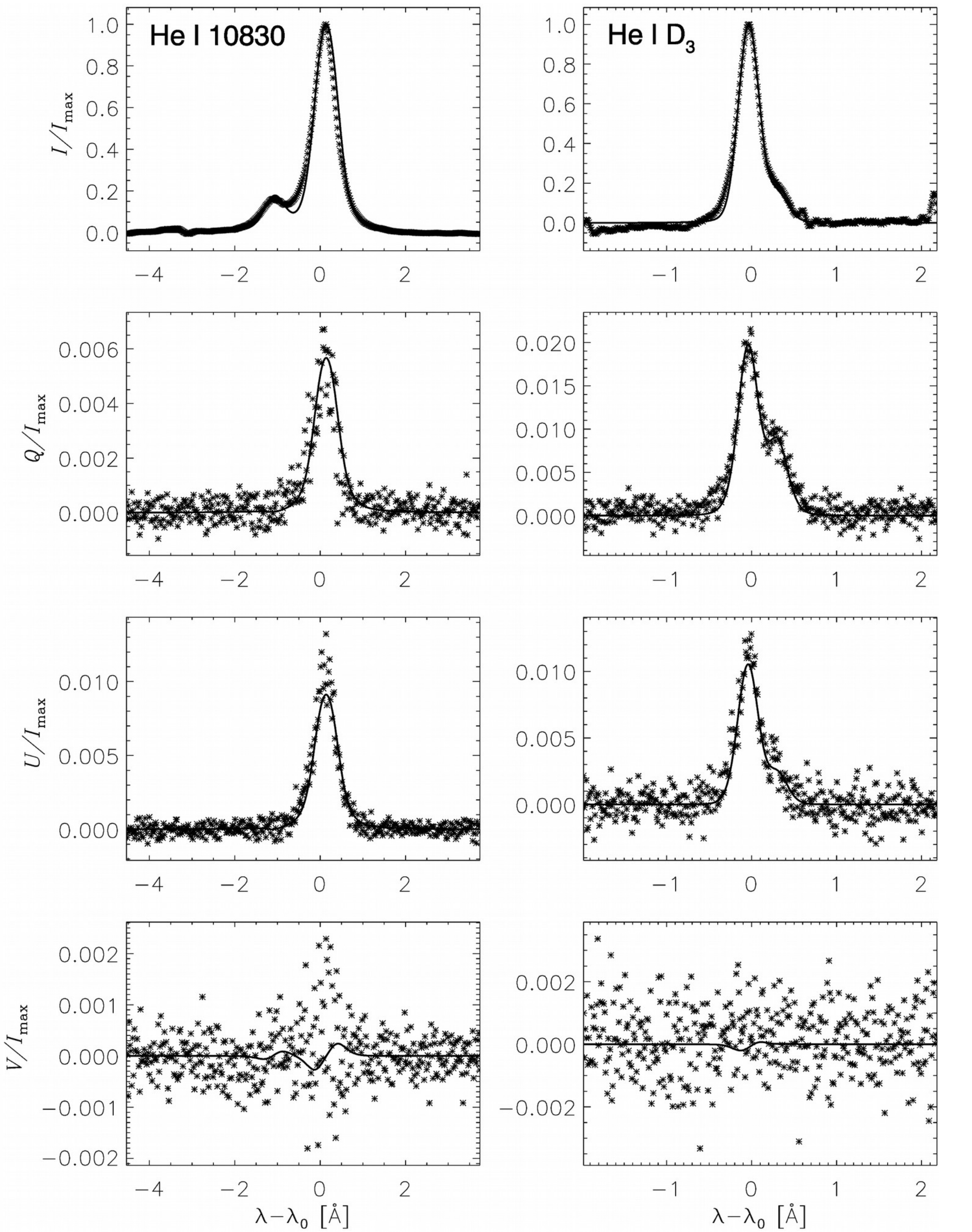


Figure 2. Multi-line inversion of simultaneous and cospatial spectro-polarimetric observations of He I 10830 (left) and D_3 (right) in a quiescent prominence, taken with THéMIS on 2007 June 29. The inverted vector magnetic field for this example is $B = 3.0$ G, $\vartheta_B = 57°.8$, and $\varphi_B = 42°.7$.

He I λ1083 nm + λ587.6 nm
Paletou et al., 29 June, 2007
@ THéMIS

PCA database:
150000 models (on the same parameter space of slide #9)

Stokes *U* and *V* of He I 1083 not quite a good fit, likely due to slightly different plasma properties for the two lines

Casini et al. 2009

Ca II 854.2 Inversion Test

“prominence” case

PCA database:
150000 models (on the same parameter space)

Synthetic “map”:
2700 random models
(~52×52 arcsec^2 map)

Inversion time:
~150 s (excluding readout of the PCA database) on one core (Intel Core2 T7600 2.33 GHz)

Parameter space

$0.01\ R_\odot \leqslant h \leqslant 0.06\ R_\odot$

off-limb ($85° \leqslant \vartheta \leqslant 95°$)

0.2 G $\leqslant B \leqslant 200$ G

(ϑ_B, φ_B) in 4π srad

10^4 K $\leqslant T \leqslant 2 \times 10^4$ K

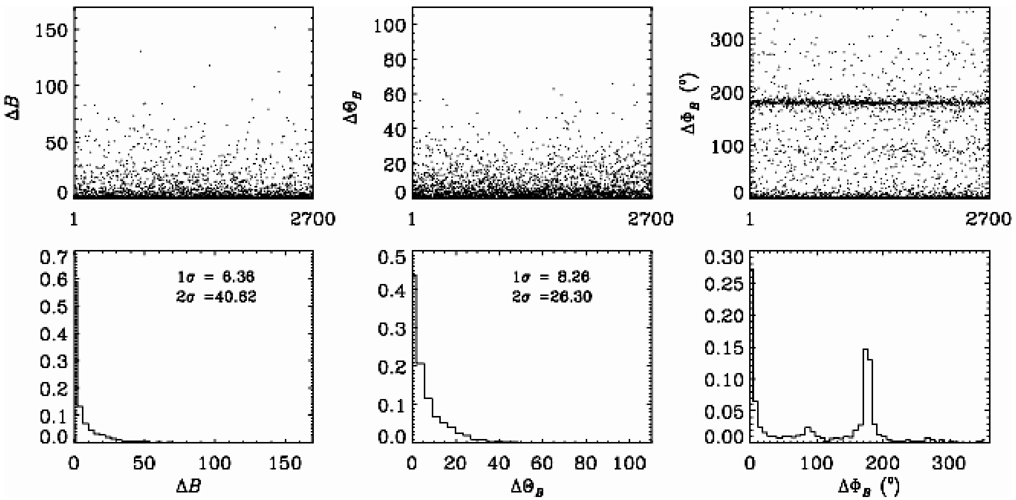
$0.5 \leqslant \tau \leqslant 1.5$

Ca II 854.2 Inversion Test

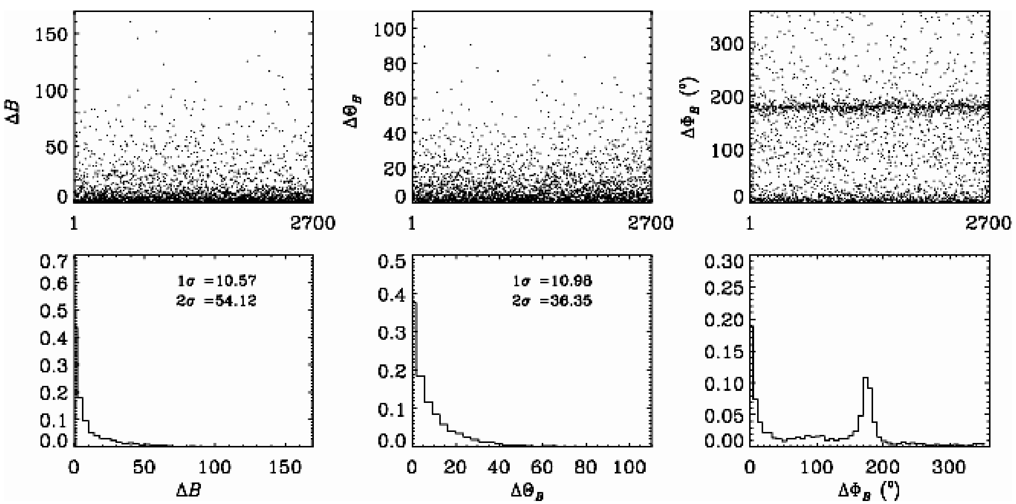
Spectro-polarimeter (ViSP)

R = 180000 (0.048 Å sampling)

No noise



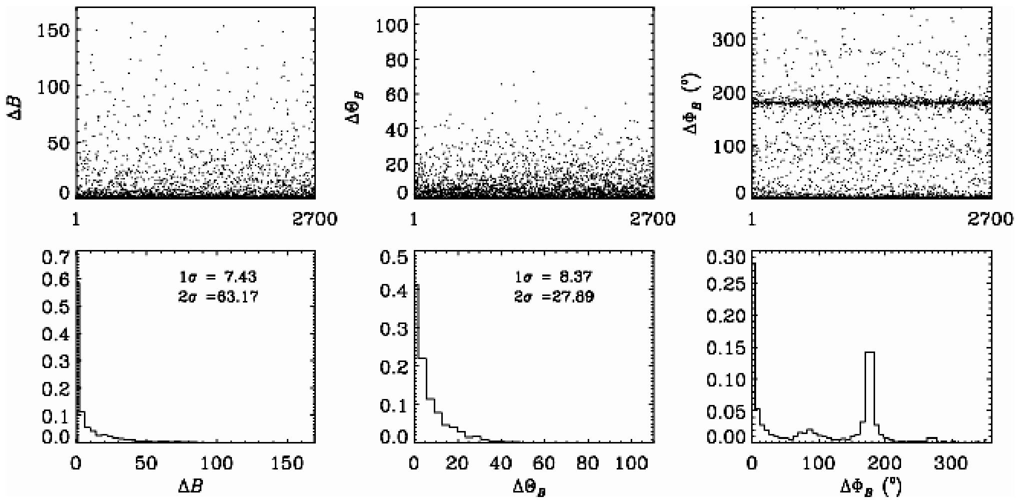
S/N = 10^3



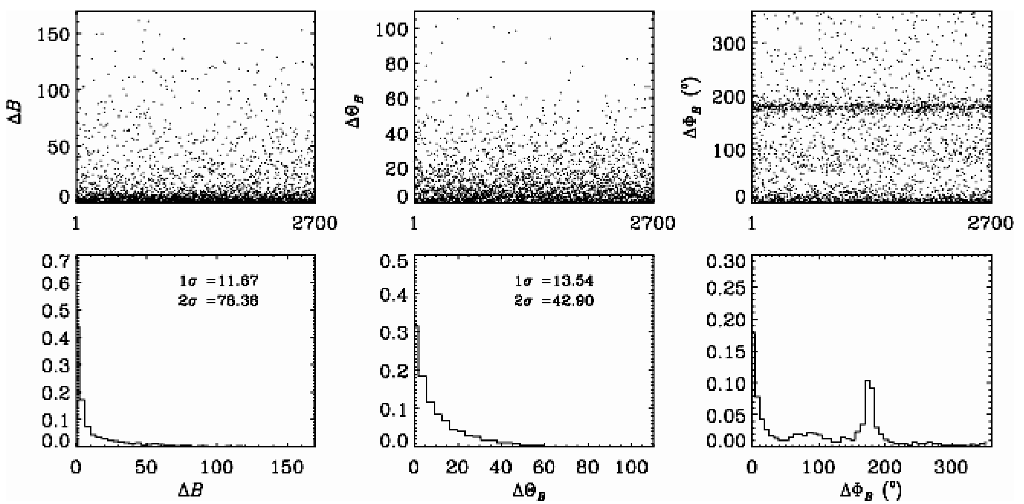
Lyot filter (ChroMag)

FWHM = 0.2 Å, 0.1 Å sampling

No noise



S/N = 10^3



Indexing of PCA Databases

MAIN IDEA

low, dominant orders of the PCA eigenprofiles capture the essential physics of line formation (higher orders are more susceptible to noise and model errors)

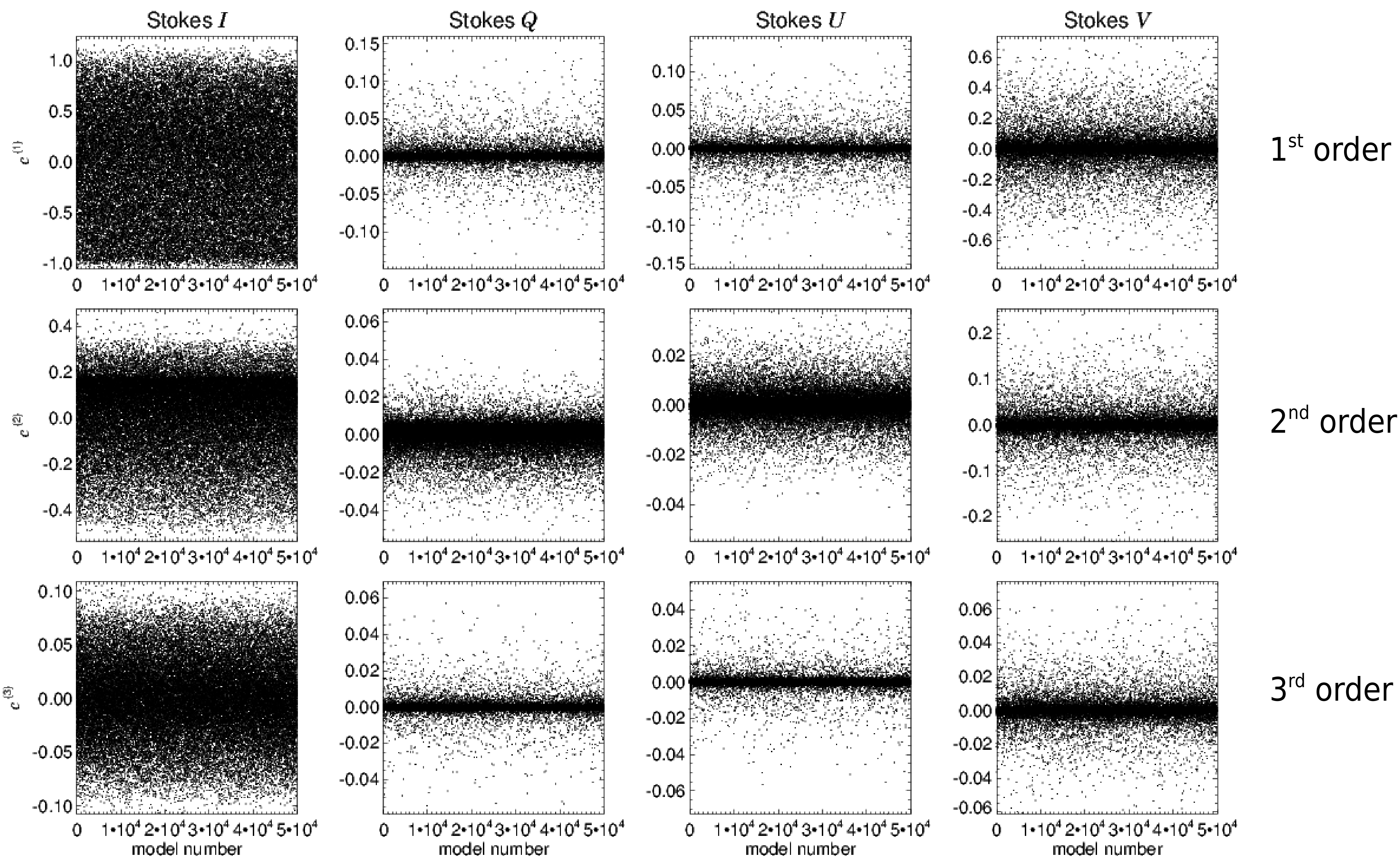
⇨ for each model realization, the ***values of the low-order PCA projections*** coarsely locate the model in the parameter space

STRATEGY

to study the ***value distribution*** of low-order PCA projections in order to partition the inversion database into ***indexed, disjoint classes***

⇨ search ***only one pertinent class per map point***, rather than the entire database, to speed up the database search

Distribution of PCA Projections
He I 1083 nm, on disk



Indexing of PCA Databases

TESTED IMPLEMENTATION

rely only on ***sign*** of PCA projection (***binary partitioning***)

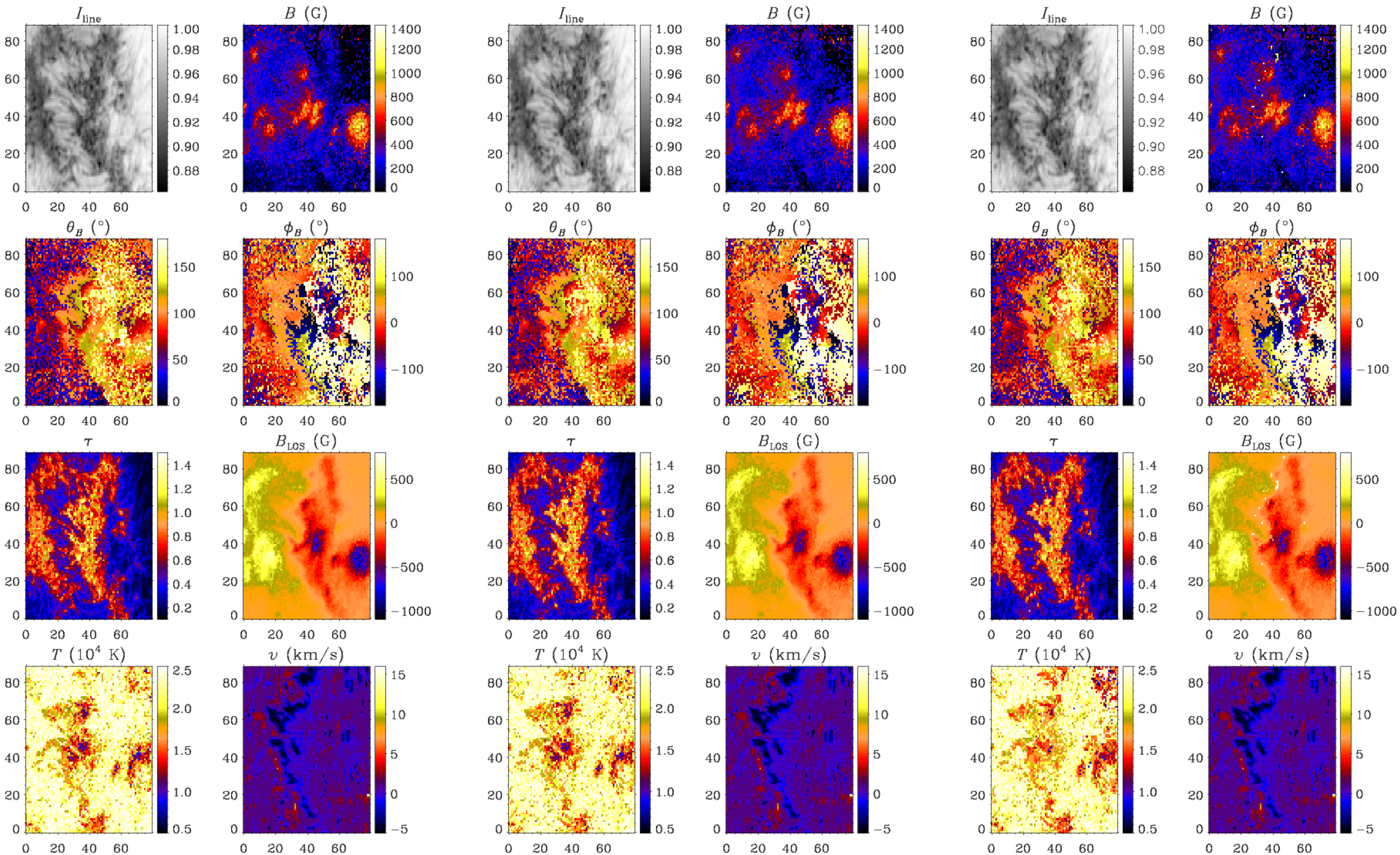
$$N_{\mathrm{classes}} = 2^{4n}$$

where n is the number of orders used for the partition

⇨ each class is identified by a ***unique binary number*** (a "bar code")

We tested such partitioning on a database of 750,000 Stokes vectors for the on-disk He I 1083 nm, and used it to invert VTT/TIP II observations by B. Lites of an A-R filament (7120 map points; single-core process)

PCA Indexed Inversion

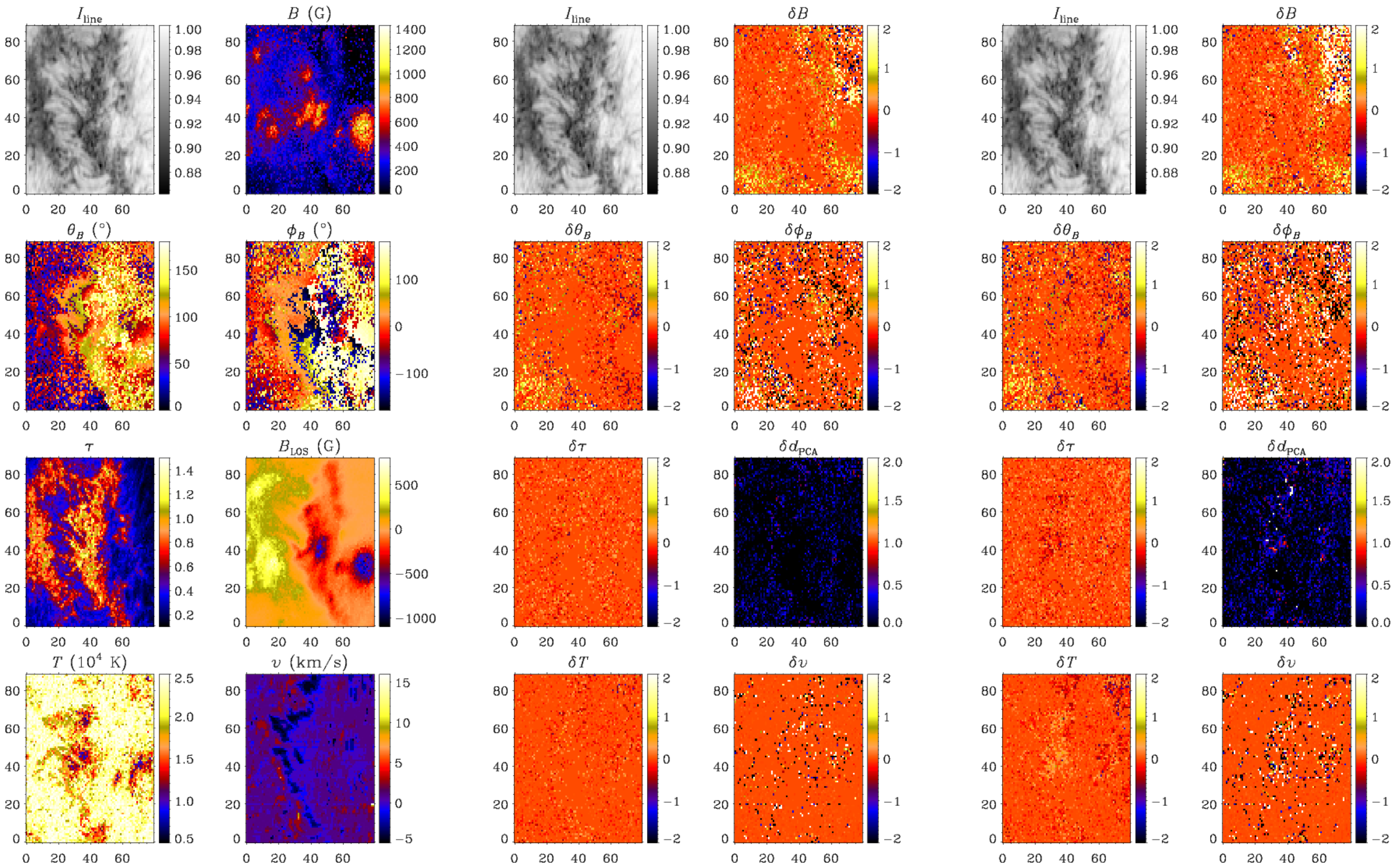


No indexing (full database; **730 s**)

1st order indexing (16 classes; **50 s**)

2nd order indexing (256 classes; **6 s**)

Difference Maps



$$\boxed{\delta p \equiv 2(p^{(k)} - p^{(0)})/(|p^{(k)}| + |p^{(0)}|)}$$

1st order indexing
(16 classes; **50 s**)

2nd order indexing
(256 classes; **6 s**)

PCA Indexed Inversion

EXAMPLE

- 10M model database
- 2 indexing orders
- full disk, 1 arcsec spatial resolution (~ 2.9M points)

⇨ 1 full inversion every ~ 9 hrs (single-core process; 2012 estimate!)

Indexing of PCA Databases

POSSIBLE DEVELOPMENT

Use both ***median and variance*** of PCA projection distributions to create a ***ternary partition*** of the inversion database

$$N_{\mathrm{classes}} = 3^{4n}$$

where n is the number of orders used for the partition

⇨ $\sim$ 80× increase in inversion speed with just one indexing order

Questions or Ideas?