Data Assimilation for Wildfire Modeling

Jonathan Beezley

January 16, 2014

Ensemble data assimilation

Purpose: Improve a model simulation by incorporating real data

- x^f: forecast (prior) model state
- x^a: analysis (posterior) model state
- y: data
- $H(x^f)$: simulated data (what the data would be if x^f represented the truth)

In a Bayesian sense, $p(x^a) = p(x^f|y) \propto p(x^f)p(y|x^f)$.
Ensemble data assimilation methods represent the distributions with matrices X^f and X^a having columns made up of perturbed model states.

Ensemble Kalman Filter (EnKF)

Uses an ensemble of solutions $\{u_k\}_{k=1}^{N}$ to estimate model errors.

Forecast step:

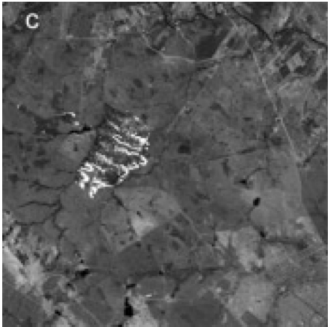
$$u_k \leftarrow \mathcal{M}(u_k)$$

Analysis step:

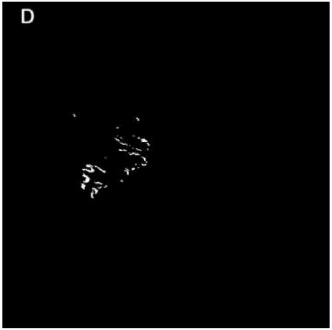
$$\begin{aligned} K_N &\leftarrow Q_N H^{\mathrm{T}} \left(H Q_N H^{\mathrm{T}} + R \right)^{-1} \\ u_k &\leftarrow u_k + K_N \left(d + e_k - H u_k \right), \quad e_k \sim \mathcal{N}(0, R) \end{aligned}$$

Real fire image data

Raw image



Post-processed image



Vodacek et al 2004

We would like to use infrared imagery from aircraft to correct errors in the fire location. Other similar data is available from satellite imagery, but at much lower resolution.

Testing the EnKF on a wildfire model

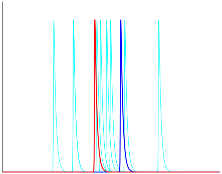
A major component of the error in wildfire modeling is determining the position of the fireline. We want to test the behavior of the EnKF under these conditions.

Test procedure:

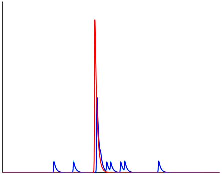
- Generate an ensemble by translating a single solution of the PDE model randomly in space.
- Use as data, a the temperature of a solution that is intentionally shifted away from the ensemble.

An example in 1D: filter degeneracy

Forecast ensemble



Analysis ensemble



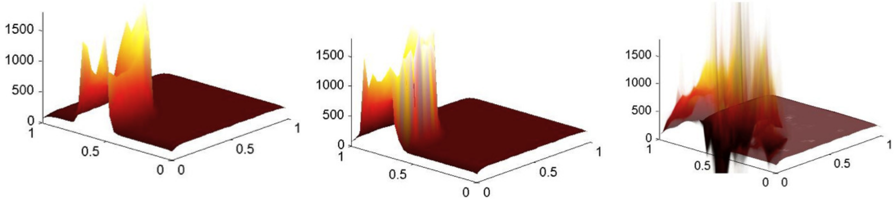
- Ensemble size, $N = 10$
- Forecast ensemble generated by translating by $N(0, \sigma^2)$, $\sigma^2 = 200\ m^2$
- Identity observation function, $H = I$
- Data covariance, $R = 10\mathrm{tr}(P^f)I$

- Cyan: ensemble
- Blue: last ensemble member
- Red: data

- $\mathrm{tr}(P^f) = 4.7 \times 10^6$
- $\mathrm{tr}(P^a) = 4.8 \times 10^2$

An example in 2D: non-physical results

Forecast ensemble — Data — Analysis ensemble



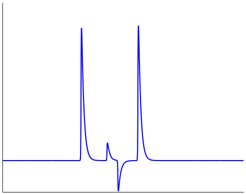
- Forecast ensemble generated by random spatial perturbations of the displayed image
- Analysis ensemble displayed as a superposition of semi-transparent images of each ensemble member
- Identity observation function, $H = I$
- Data variance, $100\ K^2$

What went wrong?

The Kalman update formula can be expressed as

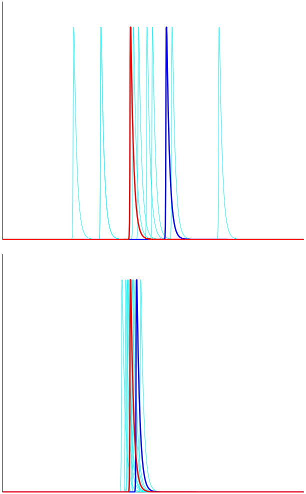
$$X^a = A(X^f)^T,$$

so $X_i^a \in \text{span}\{X^f\}$, where the analysis ensemble is made of linear combinations of the forecast.



Need to represent the position of the fire as well as its shape.

Representing spatial error, in 1D



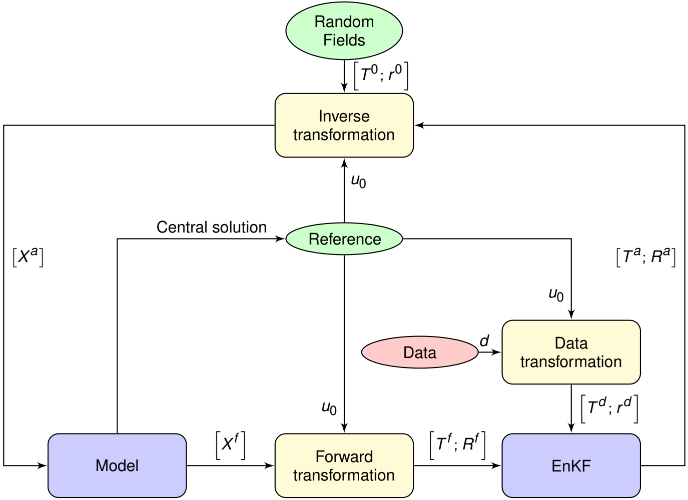
Define a non-linear transformation

$$\mathcal{T}(u_i) = \text{argmax}\{u_0\} - \text{argmax}\{u_i\} = t_i$$
$$\mathcal{T}^{-1}(t_i) = u_0(x + t_i) = u_i$$

- t_i, translation of ensemble member i from a "reference" state u_0
- run the EnKF with scalar ensemble t_i and data $\mathcal{T}(Y)$
- recover analysis ensemble by applying the inverse transformation
- $t_i \sim \mathcal{N}(m, \sigma)$, by original construction of forecast ensemble

But what about 2D?

Overview of the morphing EnKF



Morphing functions

- A morphing function, $(I + T) : \Omega \to \Omega$ defines a spatial perturbation of an image, u.
- It is invertible when $(I + T)^{-1}$ exists.
- An image u "morphed" by T is defined as $\tilde{u}(x) = u(x + Tx) = u \circ (I + T)(x)$.

$$u \qquad \circ \qquad I + T \qquad = \qquad \tilde{u}$$

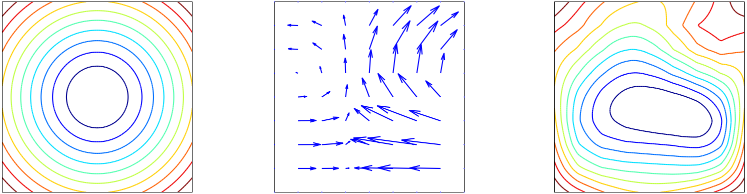


Image registration

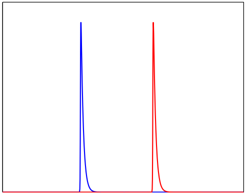
Goal: Given two images u and v, find an invertible morphing function, T, which makes $u \circ (I + T) \approx v$, while ensuring that T is "small" as possible.

Image registration problem

$$J_{u \to v}(T) = ||u \circ (I + T) - v||_{\mathcal{R}} + ||T||_{\mathcal{T}} \to \min_T$$

- $||r||_{\mathcal{R}} = c_{\mathcal{R}}||r||_2$
- $||T||_{\mathcal{T}} = c_{\mathcal{T}}||T||_2 + c_{\nabla}||\nabla T||_2$
- $c_{\mathcal{R}}$, $c_{\mathcal{T}}$, and c_{∇} are treated as optimization parameters

Minimizing the objective function



$$\nabla J_{u \to v}(0) = 0!!!$$

Problems with minimization:

- Highly nonlinear
- Many local minima
- Need an automated procedure
- Needs to be done quickly

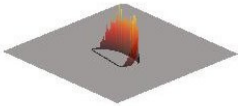
Steepest descent methods do not work in general.

Summary of the algorithm

Finding an acceptable solution requires a combination of standard techniques:

- Grid refinement: Solve the problem only approximately at first, then refine the solution on increasingly smaller sub-domains.
- Sampling: Probe the feasibility region to escape local minima.
- Image smoothing: Smooth out sharp features of the images.
- Levenberg-Marquardt: Least-square descent method designed for nonlinear problems.

u_0

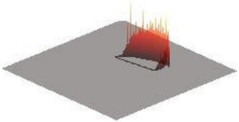


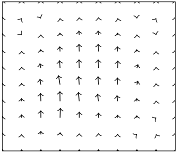
Morphing transformation

forward: $\mathcal{M}_{u_0}[u_i] = \begin{cases} T_i, & \text{from registration} \\ r_i = u_i \circ (I + T_i)^{-1} - u_0 \end{cases}$

inverse: $\mathcal{M}_{u_0}^{-1}[T_i; r_i] = (u_0 + r_i) \circ (I + T_i)$

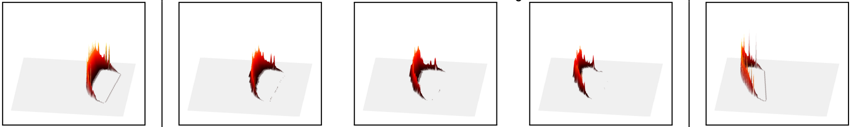
$[u_i] \quad \overset{u_0}{\longleftrightarrow} \quad [T_i \quad ; \quad r_i]$





Linear combinations of transformed states consist of a single fire moving in space.

u_0 | Intermediate states $\mathcal{M}_{u_0}^{-1}[\lambda T_i, \lambda r_i]$ | u_i

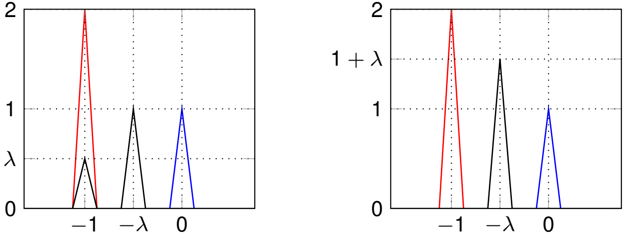


$\lambda = 0$ | $1/4$ | $1/2$ | $3/4$ | 1

Why use $r = u \circ (I+T)^{-1} - u_0$ instead of $\tilde{r} = u - u_0 \circ (I+T)$?

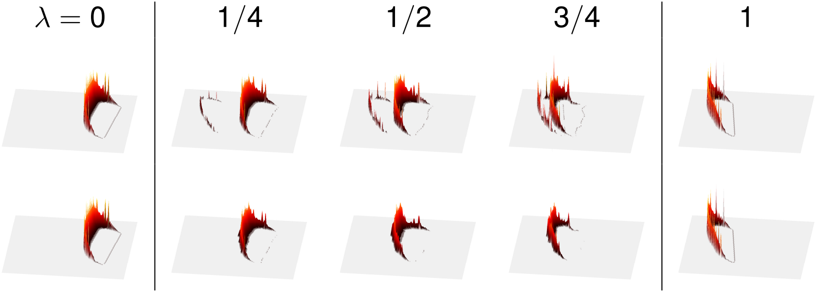
$$\tilde{u}_\lambda(x) = \varphi(x+\lambda) + \underbrace{\lambda\varphi(x+1)}_{\text{“residual”}} \qquad u_\lambda(x) = \varphi(x+\lambda) + \underbrace{\lambda\varphi(x+\lambda)}_{\text{“residual”}}$$

$$T(x) = 1$$
$$u(x) = \varphi(x)$$
$$u_0(x) = 2\varphi(x+1)$$



$\tilde{u}_\lambda$: the residual component remains fixed in space.
u_λ: the residual component moves with the image.

$$\tilde{\mathcal{M}}_{u_0}^{-1}[\lambda T, \lambda \tilde{r}] = u_0 \circ (I + \lambda T) + \overbrace{\lambda(u - u_0 \circ (I + T))}^{\text{“residual”}}$$

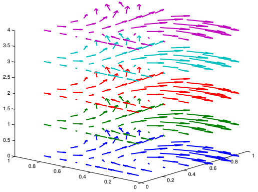


$$\mathcal{M}_{u_0}^{-1}[\lambda T, \lambda r] = u_0 \circ (I + \lambda T) + \underbrace{\lambda(u \circ (I + T)^{-1} - u_0) \circ (I + \lambda T)}_{\text{“residual”}}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \leftrightarrow \begin{bmatrix} T \\ r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} T \\ R \end{bmatrix}$$

For multi-variable states, register only one variable and use the registered morphing function to create a residual for each variable.

For $3D$ variables apply morphing function to each horizontal plane: $u_i(:,:,k)$.



Data transformation

	Original	Transformed
Data	d	$[T_d; r_d]$
Observation function	$h : \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} \mapsto [u_1]$	$\tilde{h} : \begin{bmatrix} T \\ r_1 \\ \vdots \\ r_n \end{bmatrix} \mapsto \begin{bmatrix} T \\ r_1 \end{bmatrix}$

- Data undergoes the same transformation, but only contains the registration variable.
- Observation function is modified to act on the morphed ensemble states.
- Can be extended to more general observations, but limited to "image-like" data that is a point wise function of the model variables.

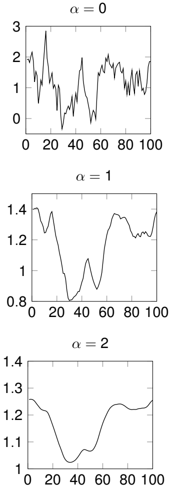
Initial ensemble

Random Fields

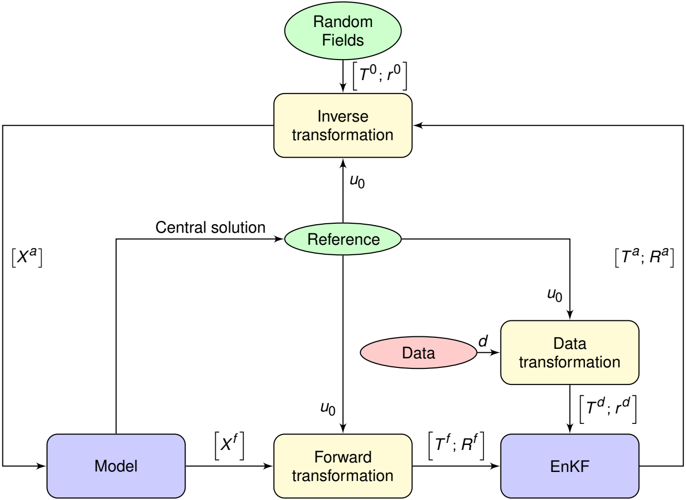
$$s_j \propto \mathcal{R}\mathrm{e}\left\{\sum_{k=0}^{n-1}(c_k + i d_k)\left(\frac{k}{h}\right)^{-1-\alpha} e^{2\pi i k j/n}\right\}$$

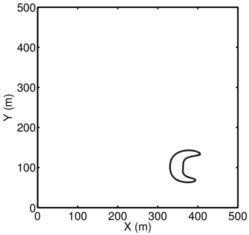
$$c_k, d_k \sim \mathcal{N}(0,1), \text{independent}$$

1. Choose a reference solution, u_0.
2. Generate random morphing function, T, out of random fields.
3. Generate random residuals, R, out of random fields for each variable.
4. $X_i^0 \leftarrow \mathcal{M}_{u_0}^{-1}([T,R])$

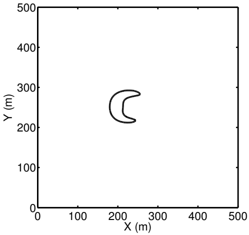


The Morphing Ensemble Kalman Filter

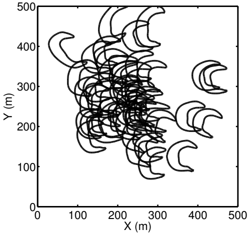




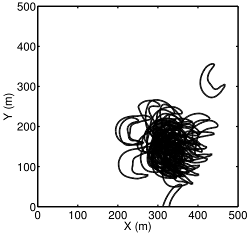
Data



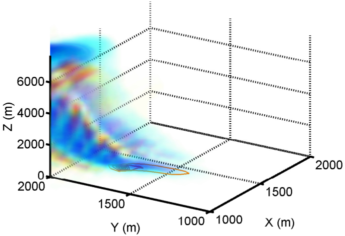
Reference



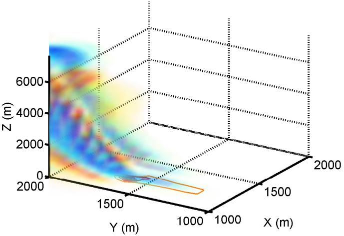
Forecast Ensemble



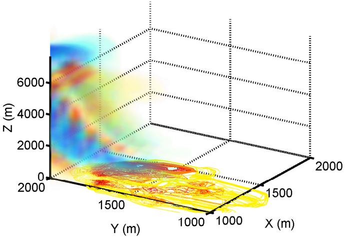
Analysis Ensemble



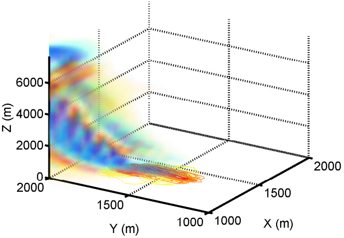
Data



Reference



Standard EnKF



Morphing EnKF