Robust Image Alignment

Oleg Makhnin
New Mexico Tech

October 14, 2013

Papers

He13: *Iterative Grassmannian Optimization for Robust Image Alignment (2013)*
by Jun He, Dejiao Zhang, Laura Balzano, Tao Tao, see
`http://arxiv.org/abs/1306.0404`

Peng12: *RASL: Robust Alignment via Sparse and Low-Rank Decomposition for Linearly Correlated Images* (2012)
by Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu and Yi Ma
`http://www.columbia.edu/~jw2966/Peng12-PAMI.pdf`

Goals:

Given several images, align them to a common template. Do it in a robust way to overcome noise, occlusions etc.

Transformed images: $D_i \circ \tau_i$, assume they align to produce a low-rank template A.
The errors E are assumed sparse.

Relaxation: $rank(A)$ is relaxed to nuclear norm (sum of singular values) and $\|E\|_0$ is relaxed to $\|E\|_1$

Examples: faces (many recognition methods need to align faces first), video processing (image stabilization, tracking objects etc.)

Formulation

Let $D = m \times n$ matrix of images (stacked as columns),
$\tau =$ set of image transformations, $D \circ \tau$: aligned images,
$A =$ common template

Peng12:

$$\min_{A,E,\tau} \|A\|_* + \lambda \|E\|_1 \qquad \text{s.t. } D \circ \tau = A + E$$

($\|\cdot\|_*$ is nuclear norm = sum of singular values), use of $\|E\|_1$ provides robustness (against occlusions + misaligned edges).

He13:

$$\min_{U,W,E,\tau} \|E\|_1 \qquad \text{s.t. } D \circ \tau = UW + E$$

and $U \in \mathcal{G}(d,m)$ where $\mathcal{G}(d,m)$ is Grassmann manifold:
$U \in \mathbb{R}^{m \times d}$ and has orthonormal columns. Dimension d is fixed.

Peng12 RASL algorithm

$$\min_{A,E,\tau} \|A\|_* + \lambda\|E\|_1 \qquad \text{s.t. } D\circ\tau = A + E$$

is a nonlinear problem: $D\circ\tau$ is nonlinear even though τ's are affine.

Solution: iterative linearization.
$D\circ(\tau+\Delta\tau) \approx D\circ\tau + \sum_{i=1}^n J_i \Delta\tau_i \varepsilon_i^T$, J_i is Jacobian of ith image w.r.t. transformation τ_i and $\{\varepsilon_i\}$ is the standard basis for $\mathbb{R}^n$.

Here, τ_i are assumed to belong to some group $\mathbb{G}$ described by p parameters (e.g. $\mathbb{G} = SE(2)$ (translations + rotations), $GL(3)$ etc), J_i is then $m \times p$, $\tau = \texttt{Stack}(\tau_i, i = 1, ..., n)$.

Solve iteratively: given τ,

$$\min_{A,E,\Delta\tau} \|A\|_* + \lambda\|E\|_1 \quad \text{s.t.} \quad D\circ\tau + \sum_{i=1}^{n} J_i \Delta\tau_i \varepsilon_i^T = A + E \quad (1)$$

is a convex problem, then set $\tau_{\texttt{new}} = \tau + \Delta\tau$.

(1) is solved using Augmented Lagrange Multiplier (ALM) Method. Let $h = D\circ\tau + \sum_{i=1}^{n} J_i \Delta\tau_i \varepsilon_i^T - A - E$. Then

$$\mathcal{L}_\mu := \|A\|_* + \lambda\|E\|_1 + \langle Y, h(A,E,\Delta\tau)\rangle + \frac{\mu}{2}\|h(A,E,\Delta\tau)\|_F^2$$

where Y is Lagrange Multiplier matrix and μ_k is an increasing sequence.

Note: $\langle X, Y\rangle = tr(X^T Y)$ and $\|\cdot\|_F$ is Frobenius norm

Then iterate

$$(A_{k+1}, E_{k+1}, \Delta\tau_{k+1}) = \arg \min_{A,E,\Delta\tau} \mathcal{L}_{\mu_k}(A, E, \Delta\tau, Y_k)$$

$$Y_{k+1} = Y_k + \mu_k h(A_{k+1}, E_{k+1}, \Delta\tau_{k+1})$$

$\min_{A,E,\Delta\tau}$ is done by alternating minimization w.r.t. A, E or $\Delta\tau$, using SVD with soft thresholding (shrinkage).
Three levels of iteration in all. Parameter λ is set to $1/\sqrt{m}$.

A MATLAB implementation is available, takes about 3 minutes on a 2.8 GHz Macbook Pro for 100 images, each 80×60.

He13 t-GRASTA algorithm

t-GRASTA = "transformed Grassmannian Robust Adaptive Subspace Tracking Algorithm"

Batch mode:

$$\min_{U,W,E,\tau} \|E\|_1 \qquad \text{s.t. } D \circ \tau = UW + E$$

is done image-wise $i = 1, ..., n$, with linearization

$$\min_{w_i,e,\Delta\tau_i} \|e\|_1 \qquad \text{s.t. } D_i \circ \tau_i + J_i \Delta\tau_i = Uw_i + e$$

interpret U as the subspace of features and w_i are weights.

Solution also involves augmented Lagrangian

$$\mathcal{L}(U, w, e, \Delta\tau, y) = \|e\|_1 + y^T h(w, e, \Delta\tau) + \frac{\mu}{2}\|h(w, e, \Delta\tau)\|_2^2,$$

$h = Uw + e - D_i \circ \tau_i - J_i \Delta\tau_i$, solved by alternating minimization...

...except for U:

U is updated in moves taking into account the geometry of Grassmannian.

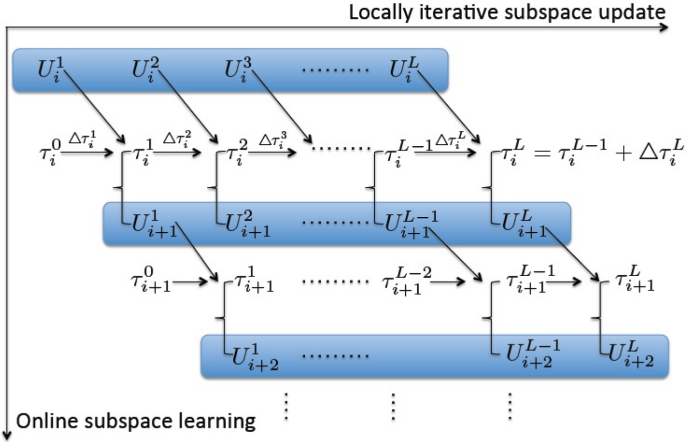
$$\frac{d\mathcal{L}}{dU} = (\lambda + \mu h(w, e, \Delta\tau))w^T$$

Then the gradient step along the geodesic in direction of $\nabla\mathcal{L} = (I - UU^T)\frac{d\mathcal{L}}{dU} =: \Gamma w^T$, which simplifies to

$$U(\eta) = U + (\cos(\eta\sigma) - 1)\frac{Uww^T}{\|w\|_2^2} - \sin(\eta\sigma)\frac{\Gamma w^T}{\|\Gamma\|\|w\|}$$

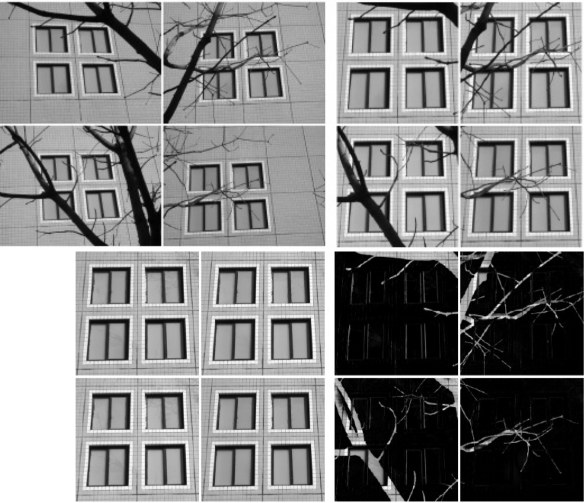
($\eta =$ step size, $\sigma := \|\Gamma\|\|w\|$ is the sole non-singular value of $\nabla\mathcal{L}$), these steps are iterated with $w = w_i$, for each image i in turn.

t-GRASTA online mode:



A "battery" of subspaces $U^\ell, \ell = 1, ..., L$ updated sequentially (exploits similarity between consecutive video images I_i, I_{i+1})

RASL example: windows



Original images (left)

Aligned images $D \circ \tau$ (right)

Low-rank component A (left)

Sparse error E (right)

t-GRASTA vs RASL comparison

(a) average of initial misaligned images; (b) average of images aligned by t-GRASTA; (c) average of background recovered by t-GRASTA; (d) average of images aligned by RASL; (e) average of background recovered by RASL.

Video (online t-GRASTA)

Row 1: original
Row 2: aligned
Row 3: low-rank component
Row 4: residual

Conclusions

*Both methods work well if the images are not too misaligned (e.g. misalignment angle up to 40°). I suspect that having large patches of the same texture helps the algorithms (as they use linear approx. of the transformed image).

* t-GRASTA is claimed about 4 times faster and takes less memory

Desirable extensions:
* non-linear/non-parametric transformations τ
* tracking multiple objects in video