Iterated Lifting for Robust Cost Optimization

Christopher Zach (christopher.zach@crl.toshiba.co.uk) — Toshiba Research Europe, Cambrige, UK

Guillaume Bourmaud (guillaume.bourmaud@u-bordeaux.fr) — University of Bordeaux, France

1) Introduction

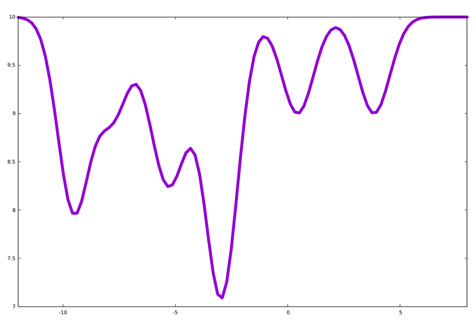
Problem statement

- Minimize a cost function involving robust data terms
- Example application: Robust bundle adjustment using the *Welsch* kernel

$$\min_{\{R_i,t_i\}_i,\{X_j\}_j} \sum_{i,j} \phi\big((\pi(R_i X_j + t_i)) - \hat{p}_{ij}\big) \text{ where } \phi_{\text{Wel},\tau}(x) = \frac{\tau^2}{2}\left(1 - e^{-\frac{x^2}{\tau^2}}\right)$$

Challenges

- large number of local minima
- large number of parameters to estimate



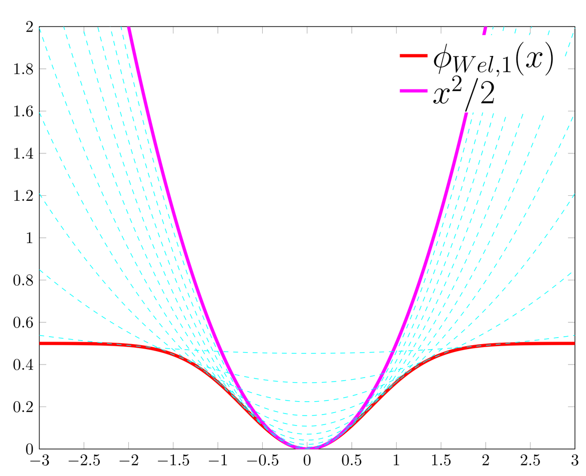
How to obtain an efficient algorithm able to escape bad local minima?

2) One solution: Half-Quadratic Lifting [1]

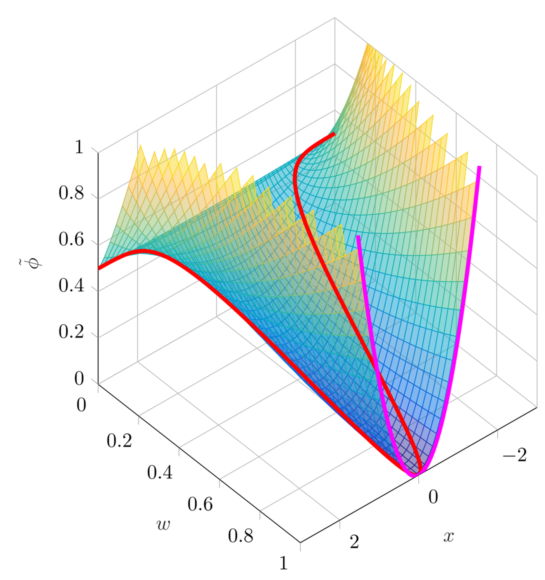
Use a *quadratic basis kernel* to lift $\phi(x)$:

$$\phi(x) = \min_{w\in[0,1]} \left\{ w\frac{x^2}{2} + \gamma(w) \right\} \stackrel{\text{def}}{=} \min_{w\in[0,1]} \tilde{\phi}(x,w)$$

Welsch kernel: $\gamma(w) = \frac{\tau^2}{2}(1 + w\log w - w)$.



Target kernel $\phi_{\text{Wel},1}(x)$ and basis kernel $\frac{x^2}{2}$



Half-quadratic lifting [1] of $\phi_{\text{Wel},1}(x)$

Half-Quadratic lifting allows utilization of efficient non-linear least-squares solvers.
Can we keep this advantage while lifting $\phi(x)$ more gradually?

3) Contributions

- Generalization of the half-quadratic lifting construction to *non-quadratic basis kernels*
- Novel technique, called *iterated lifting*, that allows to iteratively lift a robust target kernel using a less robust kernel as basis kernel
- Integrate into efficient NLSQ Levenberg-Marquardt optimizer
- Generally reaches better local minima than IRLS or half-quadratic lifting

4) Lifting using general basis kernels

Use a *non-quadratic basis kernel* $\hat{\phi}(x)$ to lift $\phi(x)$:

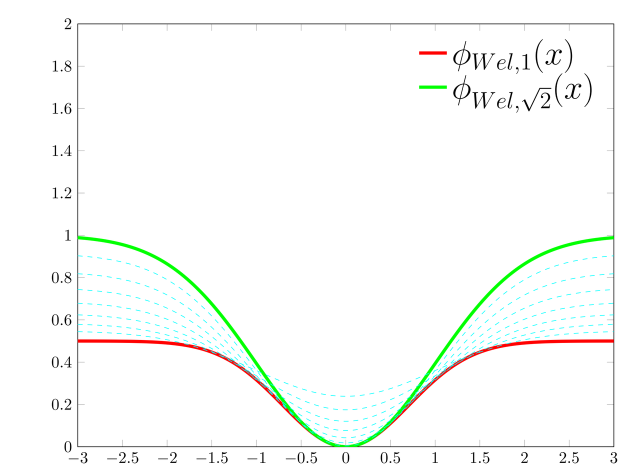
$$\phi(x) = \min_{w\in[0,1]} \left\{ w\hat{\phi}(x) + \gamma(w) \right\} \stackrel{\text{def}}{=} \min_{w\in[0,1]} \tilde{\phi}(x,w)$$

Lifting $\phi(x)$ w.r.t. a scaled version of itself is very convenient.
We have for the Welsch kernel:

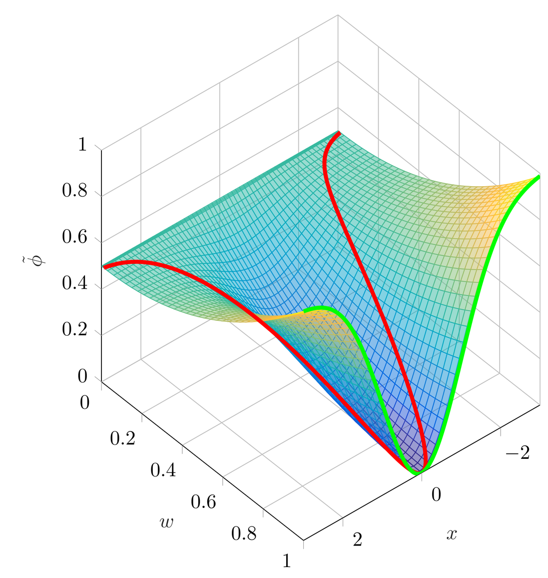
$$\phi_{\text{Wel},\tau}(x) = \min_{w\in[0,1]} w\phi_{\text{Wel},s\tau}(x) + \gamma_{\text{Wel},\tau,s}(w)$$

$$\gamma_{\text{Wel},\tau,s}(w) \stackrel{\text{def}}{=} \frac{\tau^2}{2}\left(1 + w\left(\left(s^2-1\right) w^{\frac{1}{s^2-1}} - s^2\right)\right)$$

$$\gamma_{\text{Wel},\tau,\sqrt{2}}(w) = \frac{\tau^2}{2}(w-1)^2 \qquad \gamma_{\text{Wel},\tau,\infty}(w) = \frac{\tau^2}{2}(1 + w\log w - w)$$



Target kernel $\phi_{\text{Wel},1}(x)$ and basis kernel $\phi_{\text{Wel},\sqrt{2}}(x)$



$\phi_{\text{Wel},1}(x)$ lifted w.r.t. a non-quadratic basis kernel $\phi_{\text{Wel},\sqrt{2}}(x)$

How to combine this result with efficient non-linear least-squares solvers?

5) Iterated Lifting

Idea: expand $\hat{\phi}(x)$ by its lifted representation K times.
Use $\hat{\phi}(x) = \frac{x^2}{2}$ in the final expansion to allow efficient NLSQ solvers.

Example: expand the Welsch kernel 3 times ($K = 3$, $s = \sqrt{2}$, $\mathbf{w} = (w_1, w_2, w_3)$):

$$\begin{aligned}\phi_{\text{Wel},\tau}(x) &= \min_{\mathbf{w}\ge 0} \left\{ \begin{array}{c} w_3\left(w_2\left(w_1\frac{x^2}{2} + \gamma_{\text{Wel},s^2\tau,\infty}(w_1)\right) + \gamma_{\text{Wel},s\tau,s}(w_2)\right) \\ +\gamma_{\text{Wel},\tau,s}(w_3) \end{array} \right\} \\ &= \min_{\mathbf{w}\ge 0} \left\{ \begin{array}{c} w_3\left(w_2\left(w_1\frac{x^2}{2} + 2\tau^2(1 + w_1\log w_1 - w_1)\right) + \tau^2(w_2-1)^2\right) \\ +\frac{\tau^2}{2}(w_3-1)^2 \end{array} \right\}\end{aligned}$$

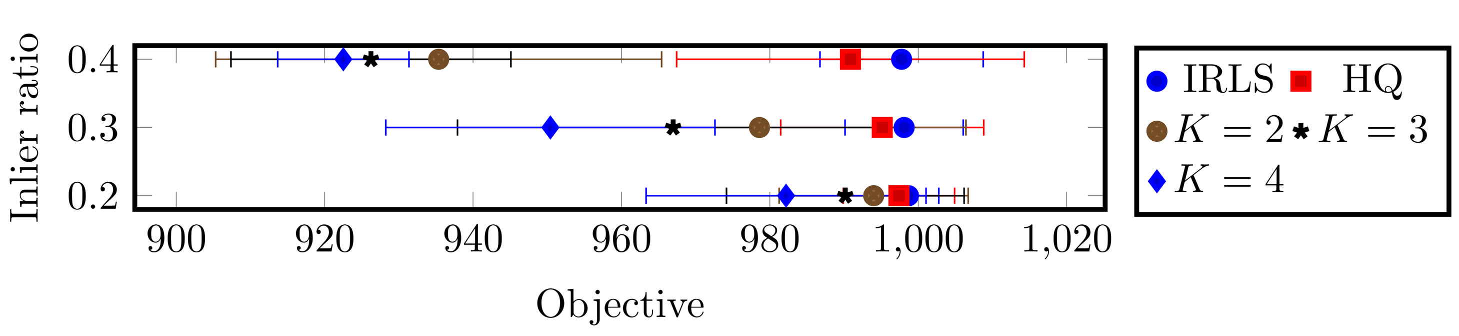
Implementation using Levenberg-Marquardt

- Parametrize $w_k = u_k^2 \ge 0$ with $u_k \in \mathbb{R}$
- Initialize all weights $w_1 \dots, w_K$ to 1 (all data points are considered inliers)
- At LM iteration $T \ge 0$ optimize over $\theta \cup \{w_1, \dots, w_{T \bmod (K+1)}\}$

6) Results

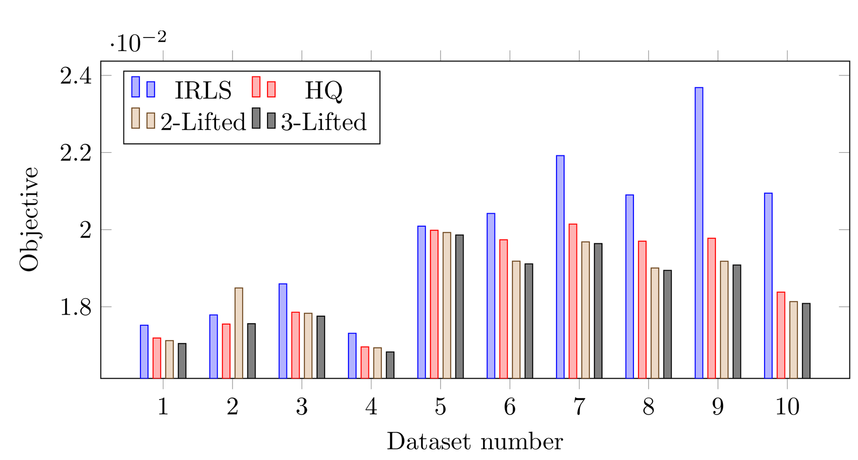
Synthetic data (random starting points)
We robustly fit a mean vector θ to synthetically generated 3-dimensional point data $\mathbf{d} = (d_1, \dots, d_{1000})$ at a given inlier ratio:

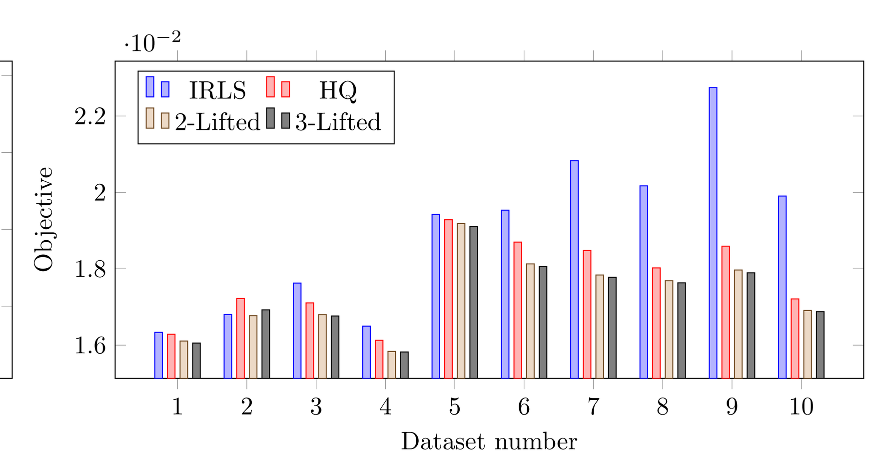
$$\min_\theta \sum_{i=1}^N \phi_{\text{Wel},\frac{1}{2}}\left(\sqrt{(d_i-\theta)^T(d_i-\theta)}\right)$$



Average objective (and standard deviation) reached by the different methods for robustly fitting the mean to synthetically generated data points.

Bundle adjustment data
We selected two problem instances from each of the 5 models in the publicly available bundle adjustment data set [2] and used the given camera parameters and 3D structure as initializer.





Objective values reached by the different methods for metric bundle adjustment and full bundle adjustment additionally optimizing over focal length and lens distortion parameters.

References

[1] Christopher Zach. Robust bundle adjustment revisited. ECCV 2014.

[2] Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. Bundle adjustment in the large. ECCV 2010.