In Perfect Shape: Certifiably Optimal 3D Shape Reconstruction from 2D Landmarks

Supplementary Material

Heng Yang and Luca Carlone
Laboratory for Information & Decision Systems (LIDS)
Massachusetts Institute of Technology
{hankyang, lcarlone}@mit.edu

1. Proof of Theorem 4

Proof. Here we prove Theorem 4 in the main document. Recall the weighted least squares optimization for shape reconstruction in eq. (11) and denote its objective function as $f(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$, with $\boldsymbol{c} = [c_1, \ldots, c_k]^\mathsf{T}$:

$$f(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t}) = \sum_{i=1}^{N} w_i \left\| \boldsymbol{z}_i - \Pi \boldsymbol{R} \left(\sum_{k=1}^{K} c_k \boldsymbol{B}_{ki} \right) - \boldsymbol{t} \right\|^2 + \alpha \sum_{k=1}^{K} |c_k| \, . \tag{A1}$$

In order to marginalize out the translation $\boldsymbol{t}$, we compute the derivative of $f(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$ w.r.t. $\boldsymbol{t}$:

$$\frac{\partial f}{\partial \boldsymbol{t}} = 2 \sum_{i=1}^{N} w_i \boldsymbol{t} - 2 \sum_{i=1}^{N} w_i \left(\boldsymbol{z}_i - \Pi \boldsymbol{R} \left(\sum_{k=1}^{K} c_k \boldsymbol{B}_{ki} \right) \right), \tag{A2}$$

and set it to $\mathbf{0}$, which allows us to write $\boldsymbol{t}^\star$ in closed form using $\boldsymbol{R}^\star$ and $\boldsymbol{c}^\star$:

$$\boldsymbol{t}^\star = \bar{\boldsymbol{z}}^w - \Pi \boldsymbol{R}^\star \left(\sum_{k=1}^{K} c_k^\star \bar{\boldsymbol{B}}_k^w \right), \tag{A3}$$

with $\bar{\boldsymbol{z}}^w$ and $\bar{\boldsymbol{B}}_k^w, k = 1, \ldots, K$, being the weighted centers of the 2D landmarks $\boldsymbol{Z}$ and the 3D basis shapes $\boldsymbol{B}_k$:

$$\bar{\boldsymbol{z}}^w = \frac{\sum_{i=1}^{N} w_i \boldsymbol{z}_i}{\sum_{i=1}^{N} w_i}, \quad \bar{\boldsymbol{B}}_k^w = \frac{\sum_{i=1}^{N} \bar{\boldsymbol{B}}_{ki}}{\sum_{i=1}^{N} w_i}. \tag{A4}$$

Then we can substitute the expression of $\boldsymbol{t}^\star$ in (A3) back into the objective function in (A1) and obtain an objective function without translation:

$$f'(\boldsymbol{c}, \boldsymbol{R}) = \sum_{i=1}^{N} w_i \left\| (\boldsymbol{z}_i - \bar{\boldsymbol{z}}^w) - \Pi \boldsymbol{R} \left(\sum_{k=1}^{K} c_i (\boldsymbol{B}_{ik} - \bar{\boldsymbol{B}}_k^w) \right) \right\|^2 + \alpha \sum_{k=1}^{K} |c_k| \tag{A5}$$

Lastly, by defining:

$$\tilde{\boldsymbol{z}}_i = \sqrt{w_i} (\boldsymbol{z}_i - \bar{\boldsymbol{z}}^w), \tag{A6}$$

$$\tilde{\boldsymbol{B}}_{ki} = \sqrt{w_i} (\boldsymbol{B}_{ki} - \bar{\boldsymbol{B}}^w), k = 1, \ldots, K \tag{A7}$$

we can see the equivalence between the objective function in eq. (A5) and the objective function in eq. (12) of Theorem 4. The constraints remain unchanged because we only marginalize out the unconstrained variable $\boldsymbol{t}$. Therefore, the shape reconstruction problem (11) is equivalent to the translation-free problem (12), and the optimal translation can be recovered using eq. (A3). □

2. Proof of Proposition 6

Proof. Here we prove the SOS relaxation of order β ($\beta \geq 2$) for the translation-free shape reconstruction problem (12) is the semidefinite program in (20). First, let us rewrite the general form of Lasserre's hierarchy of order β in eq. (8) in Theorem 3 as the following:

$$\begin{aligned} \max \quad & \gamma \\ s.t. \quad & f(\boldsymbol{x}) - \gamma = h + g, \\ & h \in \langle \boldsymbol{h} \rangle_{2\beta}, \\ & g \in Q_\beta(\boldsymbol{g}). \end{aligned} \tag{A8}$$

In words, the constraints of (A8) ask the polynomial $f(\boldsymbol{x}) - \gamma$ to be written as a sum of two polynomials h and g, with h

in the 2β-th truncated ideal of $\boldsymbol{h}$, and g in the β-th truncated quadratic module of $\boldsymbol{g}$.

Next, we use the definition of the 2β-th truncated ideal and the β-th truncated quadratic module to explicitly represent h and g. First recall the definition of the 2β-th truncated ideal in eq. (5), which states that h must be written as a sum of polynomial products between the equality constraints h_i's and the polynomial multipliers λ_i's:

$$h = \sum_{i=1}^{15} \lambda_i h_i, \tag{A9}$$

and the degree of each polynomial product $\lambda_i h_i$ must be no greater than 2β, *i.e.*, $\deg(\lambda_i h_i) \leq 2\beta$. In the translation-free shape reconstruction problem, because all the 15 equality constraints in eq. (17) (arising from $\boldsymbol{R} \in \mathrm{SO}(3)$) have degree 2, the degree of the polynomial multipliers must be at most $2\beta - 2$, *i.e.*, $\deg(\lambda_i) \leq 2\beta - 2$. Therefore, we can parametrize each λ_i using $[\boldsymbol{x}]_{2\beta-2}$, the vector of monomials up to degree $2\beta - 2$:

$$\lambda_i = \boldsymbol{\lambda}_i^{\mathsf{T}} [\boldsymbol{x}]_{2\beta-2}, \quad \boldsymbol{\lambda}_i \in \mathbb{R}^{N_\lambda}, N_\lambda = \binom{K+7+2\beta}{2\beta-2}, \tag{A10}$$

with $\boldsymbol{\lambda}_i$ being the vector of unknown coefficients associated with the monomial basis $[\boldsymbol{x}]_{2\beta-2}$. The size of $\boldsymbol{\lambda}_i$ is equal to the length of $[\boldsymbol{x}]_{2\beta-2}$, which can be computed by $\binom{n+d}{d}$, with $n = K + 9$ being the number of variables in $\boldsymbol{x}$, and $d = 2\beta - 2$ being the maximum degree of the monomial basis. Similarly for g, we recall the definition of the β-th truncated quadratic module in eq. (7), which states that g must be written as a sum of polynomial products between the inequality constraints g_k's and the SOS polynomial multipliers s_k's:

$$g = \sum_{k=0}^{2K} s_k g_k, \tag{A11}$$

and the degree of each polynomial product $s_k g_k$ must be no greater than 2β, *i.e.*, $\deg(s_k g_k) \leq 2\beta$. For our specific shape reconstruction problem, we have $g_0 := 1$, $g_k = c_k, k = 1, \ldots, K$, and $g_{K+k} = 1 - c_k^2, k = 1, \ldots, K$. Since g_0 has degree 0, s_0 can have degree up to 2β. All $g_k, k = 1, \ldots, K$, have degree 1, so $s_k, k = 1, \ldots, K$, can have degree up to $2\beta - 1$. However, because SOS polynomials can only have even degree, $s_k, k = 1, \ldots, K$ can only have degree up to $2\beta - 2$. For $g_{K+k}, k = 1, \ldots, K$, they have degree 2, so their corresponding SOS polynomial multipliers $s_{K+k}, k = 1, \ldots, K$ can have degree up to $2\beta - 2$. Now for each SOS polynomial $s_k, k = 0, \ldots, 2K$, from the Gram matrix representation in eq. (2), we can associate a PSD matrix $\boldsymbol{S}_k$ with it using corresponding monomial bases:

$$\begin{cases} s_0 = [\boldsymbol{x}]_\beta^{\mathsf{T}} \boldsymbol{S}_0 [\boldsymbol{x}]_\beta & \boldsymbol{S}_0 \in \mathcal{S}_+^{N_0}, N_0 = \binom{K+9+\beta}{\beta} \\ s_{k\neq 0} = [\boldsymbol{x}]_{\beta-1}^{\mathsf{T}} \boldsymbol{S}_k [\boldsymbol{x}]_{\beta-1} & \boldsymbol{S}_k \in \mathcal{S}_+^{N_\lambda}, N_\lambda = \binom{K+8+\beta}{\beta-1} \end{cases} \tag{A12}$$

Finally, by inserting the expressions of s_k in (A12) back to the expression of g in (A11), and inserting the expression of λ_i in (A10) back to the expression of h in (A9), we can convert the SOS relaxation of general form (A8) to the semidefinite program (20). $\square$

3. Proof of Theorem 8

Proof. According to [4], the dual SDP of (20) is the following SDP:

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & L_{\boldsymbol{y}}(f) && \text{(A13)} \\ s.t. \quad & \boldsymbol{M}_\beta(\boldsymbol{y}) \succeq 0, && \text{(A14)} \\ & \boldsymbol{M}_{\beta - v_k}(g_k \boldsymbol{y}) \succeq 0, && \text{(A15)} \\ & \boldsymbol{M}_{\beta - u_i}(h_i \boldsymbol{y}) = \boldsymbol{0}, && \text{(A16)} \\ & y_{\boldsymbol{0}} = 1, && \text{(A17)} \end{aligned}$$

where $\boldsymbol{y} \in \mathbb{R}^{N_{2\beta}}$, $N_{2\beta} = \binom{K+9+2\beta}{2\beta}$ is a vector of *moments* for a probability measure supported on $\mathcal{X}$ defined by the equalities h_i and inequalities g_k; $L_{\boldsymbol{y}}(f) = \sum_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}}$ is a linear function of $\boldsymbol{y}$, where $f_{\boldsymbol{\alpha}}$ is the coefficient of $f(\boldsymbol{x})$ associated with monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$, and $y^{\boldsymbol{\alpha}}$ is the moment of the monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$ w.r.t. the probability measure; $\boldsymbol{M}_\beta(\boldsymbol{y}) \in \mathbb{R}^{N_\beta}, N_\beta = \binom{K+9+\beta}{\beta}$ is the moment matrix of degree β that assembles all the moments in $\boldsymbol{y}$; $\boldsymbol{M}_{\beta - v_k}(g_k \boldsymbol{y}), v_k = \lceil \deg(g_k)/2 \rceil$, is the *localizing* matrix that takes some moments from the moment matrix $\boldsymbol{M}_\beta(\boldsymbol{y})$ and entry-wise multiply them with the inequality g_k (*cf.* [4] for more details); $\boldsymbol{M}_{\beta - u_i}(h_i \boldsymbol{y}), u_i = \lceil \deg(h_i)/2 \rceil$ is the localizing matrix that takes some moments from the moment matrix and entry-wise multiply them with the equality h_i. Due to strong duality of the primal-dual SDP, we have complementary slackness:

$$\boldsymbol{S}_0^{\beta\star} \boldsymbol{M}_\beta^\star = \boldsymbol{0}, \tag{A18}$$

at global optimality of the SDP pair. Since $\operatorname{corank}(\boldsymbol{S}_0^{\beta\star}) = 1$, then according to Theorem 5.7 of [4], we have $\operatorname{rank}\left(\boldsymbol{M}_\beta^\star\right) = 1$ and $f_\beta^\star$ is the global minimum of the original shape reconstruction problem (12). Further, as $\operatorname{rank}\left(\boldsymbol{M}_\beta^\star\right) = 1$, $\boldsymbol{M}_\beta^\star = \boldsymbol{v}^\star (\boldsymbol{v}^\star)^{\mathsf{T}}$ where $\boldsymbol{v}^\star = [\boldsymbol{x}^\star]_\beta$ and $\boldsymbol{x}^\star$ is the unique global minimizer of the original problem (12). However, the fact that $\boldsymbol{S}_0^{\beta\star} \boldsymbol{M}_\beta^\star = \boldsymbol{0}$ and $\boldsymbol{M}_\beta^\star = \boldsymbol{v}^\star (\boldsymbol{v}^\star)^{\mathsf{T}}$ implies:

$$\boldsymbol{S}_0^{\beta\star} \boldsymbol{v}^\star = \boldsymbol{0}, \tag{A19}$$

and $\boldsymbol{v}^\star$ is in the null-space of $\boldsymbol{S}_0^{\beta\star}$. Therefore, the solution extracted using Proposition 7 is also the unique global minimizer of problem (12). $\square$

4. Derivation of Proposition 9

Here we show the intuition for using the basis reduction in Proposition 9. In the original SOS relaxation (20), the parametrization of the SOS polynomial multipliers $s_k, k = 0, \ldots, 2K$, and the polynomial multipliers $\lambda_i, i = 1, \ldots, 15$, uses the vector of *all* monomials up to their corresponding degrees (*cf.* (A10) and (A12)), which leads to an SDP of size $N_0 = \binom{K+9+\beta}{\beta}$ that grows quadratically with the number of basis shapes K. In basis reduction, we do not limit ourselves to the vector of full monomials, but rather parametrize s_0, s_k and λ_i with unknown monomials bases $v_0[\boldsymbol{x}]$, $v_s[\boldsymbol{x}]$ and $v_\lambda[\boldsymbol{x}]$, which allows us to rewrite (21) as:

$$f(\boldsymbol{x}) - \gamma = \overbrace{v_0[\boldsymbol{x}]^\mathsf{T} \boldsymbol{S}_0 v_0[\boldsymbol{x}]}^{s_0} + \sum_{k=1}^{2K} \overbrace{\left(v_s[\boldsymbol{x}]^\mathsf{T} \boldsymbol{S}_k v_k[\boldsymbol{x}]\right)}^{s_k} g_k(\boldsymbol{x}) + \sum_{i=1}^{15} \overbrace{\left(\boldsymbol{\lambda}_i^\mathsf{T} v_\lambda[\boldsymbol{x}]\right)}^{\lambda_i} h_i(\boldsymbol{x}), \quad \text{(A20)}$$

with the hope that $v_0[\boldsymbol{x}] \subseteq [\boldsymbol{x}]_2$, $v_s[\boldsymbol{x}] \subseteq [\boldsymbol{x}]_1$ and $v_\lambda[\boldsymbol{x}] \subseteq [\boldsymbol{x}]_2$ have much smaller sizes (we limit ourselves to the case of $\beta = 2$, at which level the relaxation is empirically tight).

As described, one can see that the problem of finding smaller $v_0[\boldsymbol{x}]$, $v_s[\boldsymbol{x}]$ and $v_\lambda[\boldsymbol{x}]$, while keeping the relaxation empirically tight, is highly combinatorial in general. Therefore, our strategy is to only consider the following case:

(i) Expressive: choose $v_0[\boldsymbol{x}]$ such that s_0 contains all the monomials in $f(\boldsymbol{x}) - \gamma$,

(ii) Balanced: choose $v_s[\boldsymbol{x}]$ and $v_\lambda[\boldsymbol{x}]$ such that the sum $s_0 + \sum s_k g_k + \sum \lambda_i h_i$ can only have monomials from $f(\boldsymbol{x}) - \gamma$.

In words, condition (i) ensures that the right-hand side (RHS) of (A20) contains all the monomials of the left-hand side (LHS). Condition (ii) asks the three terms of the RHS, *i.e.*, s_0, $\sum s_k g_k$ and $\sum \lambda_i h_i$, to be self-balanced in the types of monomials. For example, if s_0 contains extra monomials that are not in the LHS, then those extra monomials better appear also in $\sum s_k g_k$ and/or $\sum \lambda_i h_i$ so that they could be canceled by summation. Under these two conditions, it is *possible* to have equation (A20) hold[1].

The choices in both conditions depend on analyzing the monomials in $f(\boldsymbol{x}) - \gamma$. Recall the expression of $f(\boldsymbol{x})$ in (12) and the expression $q_i(\boldsymbol{x})$ in (16) for each term inside the summation, it can be seen that $f(\boldsymbol{x})$ only contains the following types of monomials:

$$1 \quad \text{(A21)}$$
$$c_k, 1 \leq k \leq K \quad \text{(A22)}$$
$$c_k r_j, 1 \leq k \leq K, 1 \leq j \leq 9 \quad \text{(A23)}$$
$$c_{k_1} c_{k_2} r_{j_1} r_{j_2}, 1 \leq k_1 \leq k_2 \leq K, 1 \leq j_1 \leq j_2 \leq 9 \quad \text{(A24)}$$

and the key observation is that $f(\boldsymbol{x})$ does *not* contain degree-4 monomials *purely* in $\boldsymbol{c}$ or $\boldsymbol{r}$, *i.e.*, $c_{k_1} c_{k_2} c_{k_3} c_{k_4}$ and $r_{j_1} r_{j_2} r_{j_3} r_{j_4}$, or any degree-3 monomials in $\boldsymbol{c}$ and $\boldsymbol{r}$. Therefore, when choosing $v_0[\boldsymbol{x}]$, we can exclude degree-2 monomials purely in $\boldsymbol{c}$ and $\boldsymbol{r}$ from $[\boldsymbol{x}]_2$, and set $v_0[\boldsymbol{x}] = m_2(\boldsymbol{x}) = [1, \boldsymbol{c}^\mathsf{T}, \boldsymbol{r}^\mathsf{T}, \boldsymbol{c}^\mathsf{T} \otimes \boldsymbol{r}^\mathsf{T}]^\mathsf{T}$[2] as stated in Proposition 9. This will satisfy the expressive condition (i), because $s_0 = m_2[\boldsymbol{x}]^\mathsf{T} \boldsymbol{S}_0 m_2[\boldsymbol{x}]$ can have the following monomials:

$$1, \quad c_k, \quad c_k r_j, \quad c_{k_1} c_{k_2} r_{j_1} r_{j_2} \quad \text{(A25)}$$
$$r_j, \quad c_{k_1} c_{k_2}, \quad r_{j_1} r_{j_2}, \quad c_{k_1} c_{k_2} r_j, \quad c_k r_{j_1} r_{j_2} \quad \text{(A26)}$$

and those in (A25) cover the monomials in $f(\boldsymbol{x})$. Replacing $[\boldsymbol{x}]_2$ with $v_0[\boldsymbol{x}] = m_2(\boldsymbol{x})$ is the key step in reducing the size of the SDP, because it reduces the size of the SDP from $\binom{K+11}{2}$ to $10K + 10$, *i.e.*, from quadratic to linear in K.

In order to satisfy condition (ii), when choosing $v_s[\boldsymbol{x}]$ and $v_\lambda[\boldsymbol{x}]$, the goal is to have the product between s_k, λ_i and g_k, h_i result in monomials that appear in $f(x) - \gamma$, and ensure that monomials that do not appear in the latter can simplify our in the summation. For example, as stated in Proposition 9, we choose $v_s[\boldsymbol{x}] = [\boldsymbol{r}]_1 = [1, \boldsymbol{r}^\mathsf{T}]^\mathsf{T}$ and s_k will contain monomials 1, r_j and $r_{j_1} r_{j_2}$. Because g_k's have monomials 1, c_k and c_k^2, we can see that $\sum s_k g_k$ will contain the following monomials:

$$1, \; c_k, \; r_{j_1} r_{j_2} c_k^2, \quad \text{(A27)}$$
$$c_k^2, \; r_j, \; r_j c_k^2, \; r_{j_1} r_{j_2}, \; r_{j_1} r_{j_2} c_k, \quad \text{(A28)}$$

This still satisfies the balanced condition, because monomials of $\sum s_k g_k$ in (A27) balance with monomials of s_0 in (A25), and monomials of $\sum s_k g_k$ in (A27) balance with monomials of s_0 in (A26). Similarly, choosing $v_\lambda[\boldsymbol{x}] = [\boldsymbol{c}]_2$ makes λ_i have monomials 1, c_k and $c_{k_1} c_{k_2}$, and because h_i's have monomials 1, r_j and $r_{j_1} r_{j_2}$, we see that $\sum \lambda_i h_i$ contains the following monomials:

$$1, \; c_k, \; c_k r_j, \; c_{k_1} c_{k_2} r_{j_1} r_{j_2}, \quad \text{(A29)}$$
$$r_j, \; r_{j_1} r_{j_2}, \; c_k r_{j_1} r_{j_2}, \; c_{k_1} c_{k_2}, \; c_{k_1} c_{k_2} r_j, \quad \text{(A30)}$$

which balance with monomials in s_0 from (A25) and (A26).

We remark that we cannot guarantee that the SOS relaxation resulting from basis reduction can achieve the same performance as the original SOS relaxation and we cannot guarantee our choice of basis is "optimal" in any sense.

[1] Whether or not these are sufficient or necessary conditions remains open. However, leveraging Theorem 8 we can still check optimality a posteriori.

[2] A more rigorous analysis should follow the rules of Newton Polytope [3], but the intuition is the same as what we describe here.

Therefore, in practice, one needs to check the solution and compute corank($\boldsymbol{S}_0^{2\star}$) and η_2 to check the optimality of the solution produced by (25). Moreover, it remains an open problem to find a better set of monomials bases to achieve better reduction (*e.g.*, knowing more about the algebraic geometry of g_k and h_i could possibly enable using the *standard monomials* as a set of bases [3]).

5. Derivation of Algorithm 1

For a complete discussion of graduated non-convexity and its applications for robust spatial perception, please see [6].

In the main document, for robust shape reconstruction, we adopt the TLS shape reconstruction formulation:

$$\min_{\substack{c_k \geq 0,\\ k=1,\dots,K\\ \boldsymbol{t}\in\mathbb{R}^2, \boldsymbol{R}\in\mathrm{SO}(3)}} \sum_{i=1}^{N} \rho_{\bar{c}}\left(r_i(c_k, \boldsymbol{R}, \boldsymbol{t})\right) + \alpha \sum_{k=1}^{K} c_k \tag{A31}$$

where $r_i(c_k, \boldsymbol{R}, \boldsymbol{t}) := \left\| \boldsymbol{z}_i - \Pi \boldsymbol{R}\left(\sum_{k=1}^{K} c_k \boldsymbol{B}_{ki}\right) - \boldsymbol{t} \right\|$ is called the *residual*, and $\rho_{\bar{c}}(r) = \min(r^2, \bar{c}^2)$ implements a truncated least squares cost. Recalling that $\rho_{\bar{c}}(r) = \min(r^2, \bar{c}^2) = \min_{w\in\{0,1\}} wr^2 + (1-w)\bar{c}^2$, we can rewrite the TLS shape reconstruction as a joint optimization of $(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$ and the binary variables w_i's, as in eq. (28) in the main document. However, as hinted in the main document, due to the non-convexity of the TLS cost, directly solving the joint problem or alternating between solving for $(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$ and binary variables w_i's would require an initial guess and is prone to bad local optima.

The idea of graduated non-convexity (GNC) [2] is to introduce a surrogate function $\rho_{\bar{c}}^{\mu}(r)$, governed by a control parameter μ, such that changing μ allows $\rho_{\bar{c}}^{\mu}(r)$ to start from a convex proxy of $\rho_{\bar{c}}(r)$, and gradually increase the amount of non-convexity till the original TLS function $\rho_{\bar{c}}(r)$ is recovered. The surrogate function for TLS is stated below.

Proposition A1 (Truncated Least Squares (TLS) and GNC)**.** *The truncated least squares function is defined as:*

$$\rho_{\bar{c}}(r) = \begin{cases} r^2 & \text{if } r^2 \in [0, \bar{c}^2] \\ \bar{c}^2 & \text{if } r^2 \in [\bar{c}^2, +\infty) \end{cases}, \tag{A32}$$

where $\bar{c}$ is a given truncation threshold. The GNC surrogate function with control parameter μ is:

$$\rho_{\bar{c}}^{\mu}(r) = \begin{cases} r^2 & \text{if } r^2 \in \left[0, \frac{\mu}{\mu+1}\bar{c}^2\right] \\ 2\bar{c}|r|\sqrt{\mu(\mu+1)} - \mu(\bar{c}^2 + r^2) & \text{if } r^2 \in \left[\frac{\mu}{\mu+1}\bar{c}^2, \frac{\mu+1}{\mu}\bar{c}^2\right] \\ \bar{c}^2 & \text{if } r^2 \in \left[\frac{\mu+1}{\mu}\bar{c}^2, +\infty\right) \end{cases} \tag{A33}$$

By inspection, one can verify $\rho_{\bar{c}}^{\mu}(r)$ is convex for μ approaching zero ($(\rho_{\bar{c}}^{\mu}(r))'' = -2\mu \to 0$) and retrieves $\rho_{\bar{c}}(r)$ in (A32) for $\mu \to +\infty$. An illustration of $\rho_{\bar{c}}^{\mu}(r)$ is given in Fig. A1.

The nice property of the GNC surrogate function is that when μ is close to zero, $\rho_{\bar{c}}^{\mu}$ is convex, which means the only non-convexity of problem (A31) comes from the constraints and can be relaxed using the SOS relaxations.

For the GNC surrogate function $\rho_{\bar{c}}^{\mu}$, the simple trick of introducing binary variables ($\rho_{\bar{c}}(r) = \min_{w\in\{0,1\}} wr^2 + (1-w)\bar{c}^2$) would not work. However, Black and Rangarajan [1] showed that this idea of introducing an *outlier variable*[3] can be generalized to many robust cost functions. In particular, for the GNC surrogate function, we have the following.

Theorem A2 (Black-Rangarajan Duality for GNC surrogate TLS)**.** *The GNC surrogate TLS shape reconstruction:*

$$\min_{\substack{c_k \geq 0,\\ k=1,\dots,K\\ \boldsymbol{t}\in\mathbb{R}^2, \boldsymbol{R}\in\mathrm{SO}(3)}} \sum_{i=1}^{N} \rho_{\bar{c}}^{\mu}\left(r_i(c_k, \boldsymbol{R}, \boldsymbol{t})\right) + \alpha \sum_{k=1}^{K} c_k \tag{A34}$$

with $\rho_{\bar{c}}^{\mu}(r)$ defined in (A33), is equivalent to the following optimization with outlier variables w_i's:

$$\min_{\substack{c_k \geq 0,\\ k=1,\dots,K\\ \boldsymbol{t}\in\mathbb{R}^2, \boldsymbol{R}\in\mathrm{SO}(3)\\ w_i\in[0,1], i=1,\dots,N}} \sum_{i=1}^{N} \left[w_i r_i^2(c_k, \boldsymbol{R}, \boldsymbol{t}) + \Phi_{\bar{c}}^{\mu}(w_i) \right] + \alpha \sum_{k=1}^{K} c_k \tag{A35}$$

where $\Phi_{\bar{c}}^{\mu}(w_i)$ is the following outlier process:

$$\Phi_{\bar{c}}^{\mu}(w_i) = \frac{\mu(1-w_i)}{\mu + w_i}\bar{c}^2. \tag{A36}$$

Proof. The derivation of $\Phi_{\bar{c}}^{\mu}(w_i)$ in (A36) follows the Black-Rangarajan procedure in Fig. 10 of [1]. $\square$

In words, the Black-Rangarajan duality allows us to rewrite the non-convex shape reconstruction problem as a joint optimization in $(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$ and outlier variables w_i's. The interested readers can find closed-form outlier processes for many other robust cost functions in the original paper [1].

Leveraging the Black-Rangarajan duality, for any given choice of the control parameter μ, we can solve problem (A35) in two steps: first we solve $(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$ using Shape* with fixed weights w_i's, and then we update the weights with fixed $(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$. In particular, at each iteration τ (corresponding to a given control parameter μ), we perform the following:

1. **Variable update**: minimize (A35) with respect to

[3] w can be thought of as an outlier variable: when $w = 1$, the measurement is an inlier, when $w = 1$, the measurement is an outlier.

$(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$, with fixed weights $w_i^{(\tau-1)}$:

$$
c_k^{(\tau)}, \boldsymbol{R}^{(\tau)}, \boldsymbol{t}^{(\tau)} = \underset{\substack{c_k \geq 0, \\ k=1,\dots,K \\ \boldsymbol{t} \in \mathbb{R}^2, \boldsymbol{R} \in \mathrm{SO}(3)}}{\arg\min} \sum_{i=1}^{N} w_i^{(\tau-1)} r_i^2(c_k, \boldsymbol{R}, \boldsymbol{t}) + \alpha \sum_{k=1}^{K} c_k, \quad \text{(A37)}
$$

where we have dropped the term $\sum_{i=1}^{N} \Phi_{\bar{c}}^{\mu}(w_i)$ because it is independent from $(\boldsymbol{c}, \boldsymbol{R}, \boldsymbol{t})$. This problem is exactly the weighted least squares problem (11) and can be solved using Shape* (*cf.* line 4 in Algorithm 1). Using the solutions $(c_k^{(\tau)}, \boldsymbol{R}^{(\tau)}, \boldsymbol{t}^{(\tau)})$, we can compute the residuals $r_i^{(\tau)}$ (*cf.* line 5 in Algorithm 1).

2. **Weight update**: minimize (A35) with respect to w_i, with fixed residuals $r_i^{(\tau)}$:

$$
w_i^{(\tau)} = \underset{w_i \in [0,1], i=1,\dots,N}{\arg\min} \sum_{i=1}^{N} (r_i^{(\tau)})^2 w_i + \Phi_{\bar{c}}^{\mu}(w_i), \quad \text{(A38)}
$$

where we have dropped $\sum_{k=1}^{K} c_k^{(\tau)}$ because it is a constant for the optimization. This optimization, fortunately, can be solved in closed-form. We take the gradient of the objective function with respect to w_i:

$$
\begin{aligned}
\nabla_{w_i} &= (r_i^{(\tau)})^2 + \nabla_{w_i} \Phi_{\bar{c}}^{\mu}(w_i) \\
&= (r_i^{(\tau)})^2 - \tfrac{\mu(\mu+1)}{(\mu+w_i)^2} \bar{c}^2 \qquad \text{(A39)}
\end{aligned}
$$

and observe that $\nabla_{w_i} = (r_i^{(\tau)})^2 - \frac{\mu+1}{\mu}\bar{c}^2$ when $w_i = 0$, and $\nabla_{w_i} = (r_i^{(\tau)})^2 - \frac{\mu}{\mu+1}\bar{c}^2$ when $w_i = 1$. Therefore, the global minimizer $w_i^{\star} := w_i^{(\tau)}$ is:

$$
w_i^{(\tau)} = \begin{cases} 0 & \text{if } (r_i^{(\tau)})^2 \in \left[\frac{\mu+1}{\mu}\bar{c}^2, +\infty\right] \\ \frac{\bar{c}}{r_i^{(\tau)}} \sqrt{\mu(\mu+1)} - \mu & \text{if } (r_i^{(\tau)})^2 \in \left[\frac{\mu}{\mu+1}\bar{c}^2, \frac{\mu+1}{\mu}\bar{c}^2\right] \\ 1 & \text{if } (r_i^{(\tau)})^2 \in \left[0, \frac{\mu}{\mu+1}\bar{c}^2\right] \end{cases} . \quad \text{(A40)}
$$

and this is the weight update rule in line 6 of Algorithm 1.

After both the variables and weights are updated using the two-stage approach described above, we increase the control parameter μ to increase the non-convexity of the surrogate function $\rho_{\bar{c}}^{\mu}$ (*cf.* line 10 of Algorithm 1). At the next iteration $\tau + 1$, the updated weights are used to perform the variable update. The iterations terminate when the change in the objective function becomes negligible (*cf.* line 8 of Algorithm 1) or after a maximum number of iterations (*cf.* line 3 of Algorithm 1). Note that all weights are initialized to 1 (*cf.* line 1 in Algorithm 1), which means that initially all measurements are tentatively accepted as inliers, therefore no prior information about inlier/outlier is required.

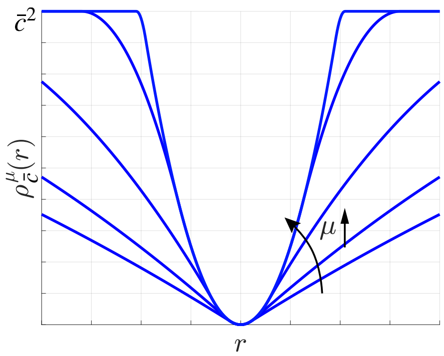


Figure A1: Graduated Non-Convexity (GNC) with control parameter μ for the Truncated Least Squares (TLS) cost.

6. FG3DCar Qualitative Results

Fig. A2 shows 9 full qualitative results comparing the performances of Altern+Robust [7], Convex+Robust [7] and Shape# on the FG3DCar [5] dataset under 10% to 70% outlier rates. One can further see that the performance of Shape# is sensitive to 70% outliers, while the performances of Altern+Robust and Convex+Robust gradually degrade and fail at 50% to 60% outliers.

Figure A2: Qualitative results on the FG3DCar dataset [5] under $10 - 70\%$ outlier rates using Altern+Robust [7], Convex+Robust [7], and Shape#. Yellow: shape reconstruction result projected onto the image. Green: inliers. Red: outliers. Circle: 3D landmark. Square: 2D landmark. [Best viewed electronically.]

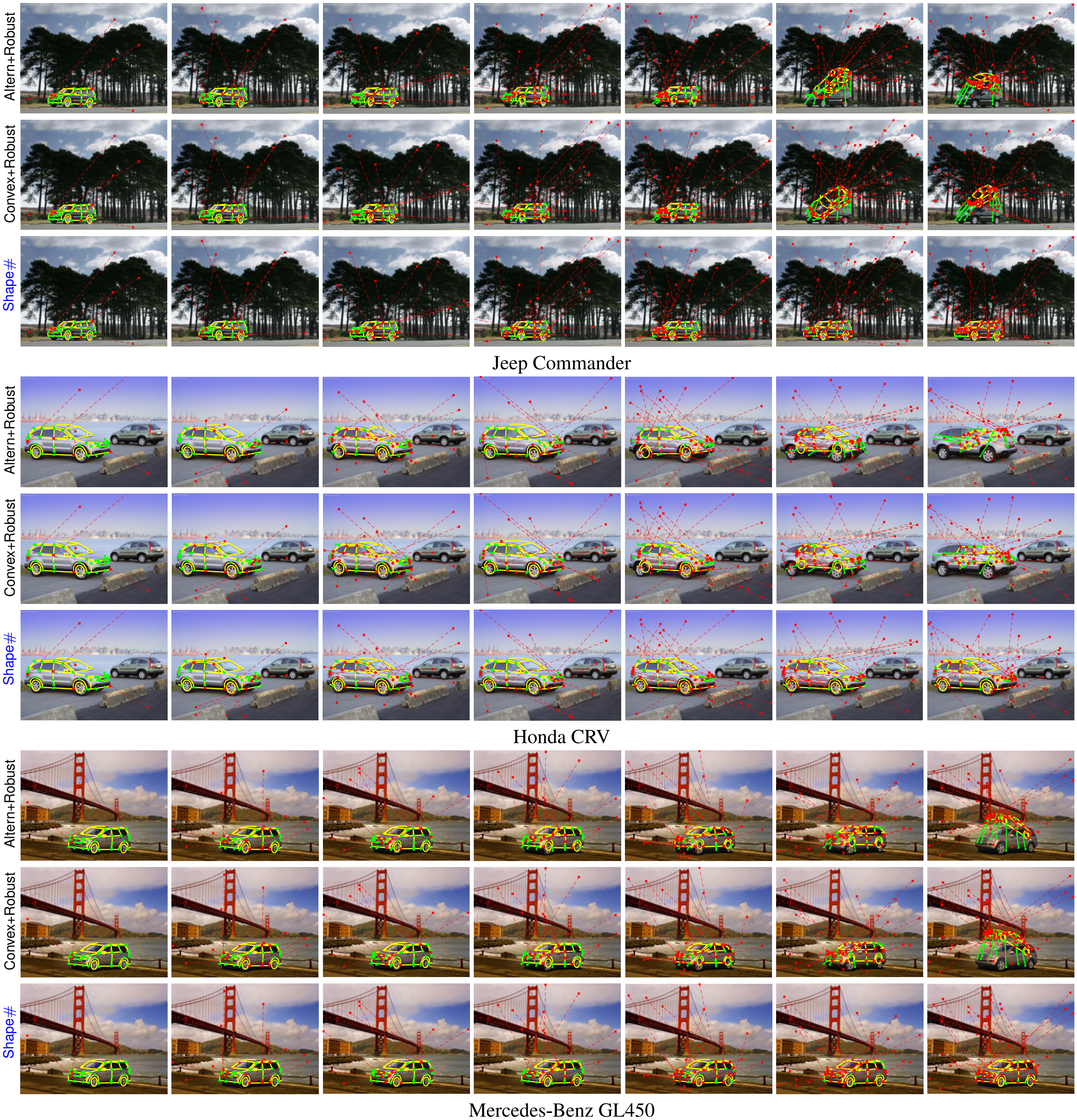
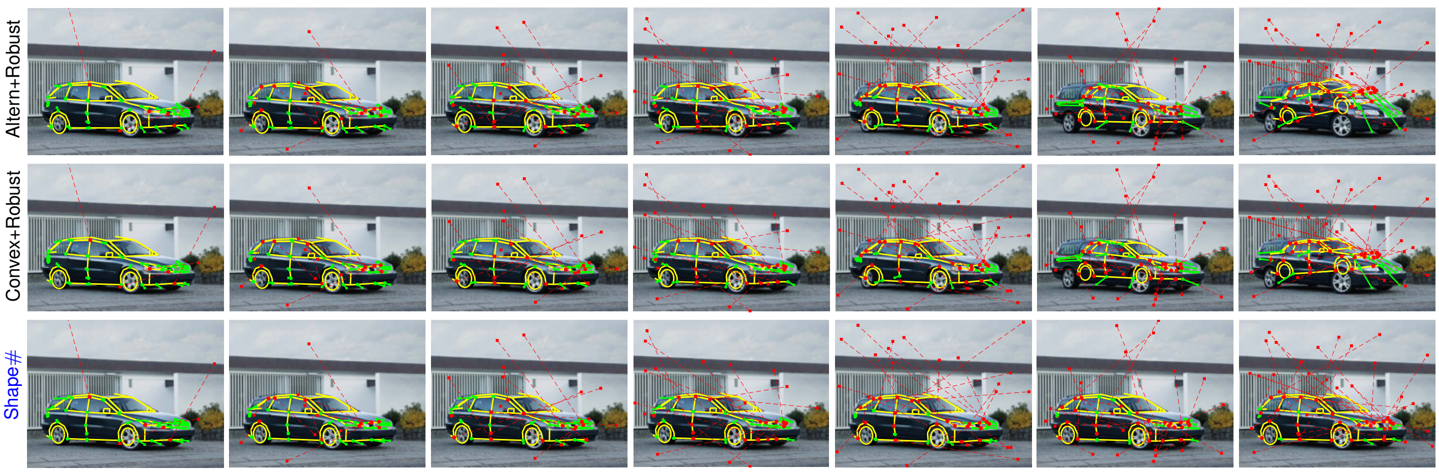
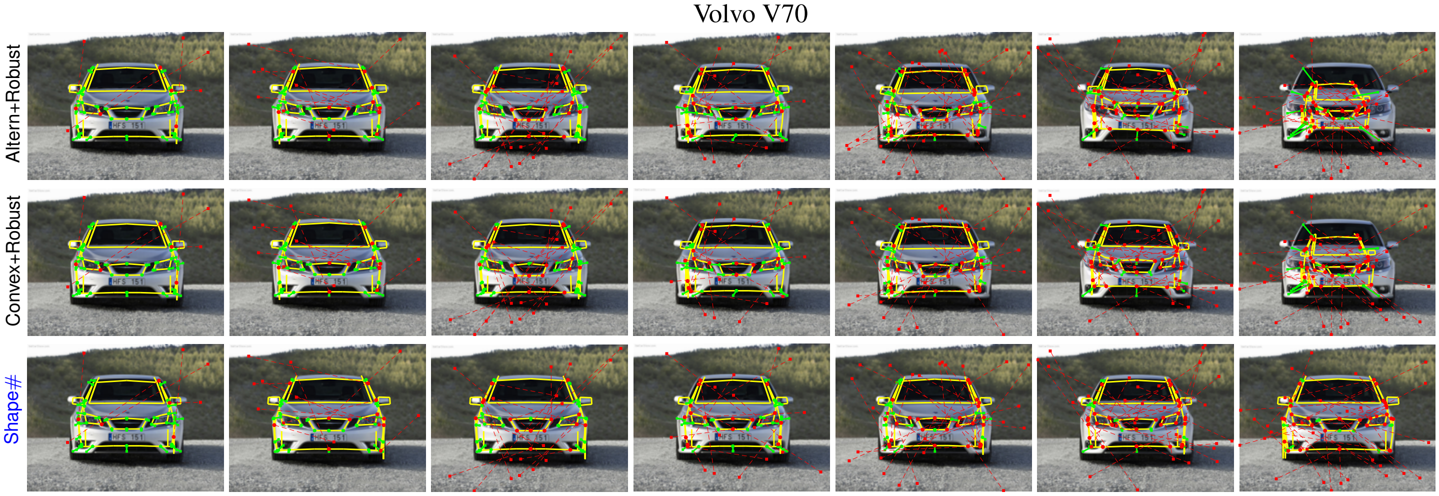


Figure A2: Qualitative results on the FG3DCar dataset [5] under $10-70\%$ outlier rates using Altern+Robust [7], Convex+Robust [7], and Shape#. Yellow: shape reconstruction result projected onto the image. Green: inliers. Red: outliers. Circle: 3D landmark. Square: 2D landmark. (cont.) [Best viewed electronically.]

Vauxhall Zafira



Volvo V70



Saab 93

Figure A2: Qualitative results on the FG3DCar dataset [5] under $10 - 70\%$ outlier rates using Altern+Robust [7], Convex+Robust [7], and Shape#. Yellow: shape reconstruction result projected onto the image. Green: inliers. Red: outliers. Circle: 3D landmark. Square: 2D landmark. (cont.) [Best viewed electronically.]

References

[1] Michael J. Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Intl. J. of Computer Vision*, 19(1):57–91, 1996. 4

[2] Andrew Blake and Andrew Zisserman. *Visual reconstruction*. MIT Press, 1987. 4

[3] Grigoriy Blekherman, Pablo A Parrilo, and Rekha R Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012. 3, 4

[4] Jean-Bernard Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2010. 2

[5] Yen-Liang Lin, Vlad I. Morariu, Winston H. Hsu, and Larry S. Davis. Jointly optimizing 3D model fitting and fine-grained classification. In *European Conf. on Computer Vision (ECCV)*, 2014. 5, 6, 7, 8

[6] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters (RA-L)*, 2020. 4

[7] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, and Kostas Daniilidis. Sparse representation for 3D shape estimation: A convex relaxation approach. *IEEE Trans. Pattern Anal. Machine Intell.*, 39(8):1648–1661, 2017. 5, 6, 7, 8