Improved Search Strategies for Determining Facial Expression

Michael Bao[1,2]   Xinru Hua[1]   Ronald Fedkiw[1,2]
[1]Stanford University   [2]Industrial Light & Magic
[1]{mikebao,huaxinru,rfedkiw}@stanford.edu

Abstract

It is well known that popular optimization techniques can lead to overfitting or even a lack of convergence altogether; thus, practitioners often utilize ad hoc regularization terms added to the energy functional. When carefully crafted, these regularizations can produce compelling results. However, regularization changes both the energy landscape and the solution to the optimization problem, which can result in underfitting. Surprisingly, many practitioners both add regularization and claim that their model lacks the expressivity to fit the data. Motivated by a geometric interpretation of the linearized search space, we propose an approach that ameliorates overfitting without the need for regularization terms that restrict the expressiveness of the underlying model. We illustrate the efficacy of our approach on minimization problems related to three-dimensional facial expression estimation where overfitting clouds semantic understanding and regularization may lead to underfitting that misses or misinterprets subtle expressions.

1. Introduction

There exist many standard methods for finding the solution to an unconstrained nonlinear optimization problem; however, generally speaking, one typically approaches these problems by iteratively solving a sequence of linear problems aiming to progress towards the solution of the fully nonlinear problem. For example, the Gauss-Newton method [25] aims to minimize an energy function $f(x)$ by iteratively solving a linear problem $A^T A \delta x = A^T b$ where A is the Jacobian of f evaluated at the current parameter state x, b is the current residual, and δx is the desired step in parameter values to move towards the solution. This process is repeated until convergence. Although this linear problem minimizes $\|A\delta x - b\|_2$, it is unclear in practice that minimizing this L2 norm is the best way to make progress towards the value of x that best minimizes $f(x)$. In fact, the Levenberg-Marquardt and the Dogleg trust-region methods blatantly modify the least squares problem claiming to make the Gauss-Newton method more robust [24, 26].

Even when one can trust the linear problem to provide progress towards the solution or can modify the linear problem to be more trust-worthy, one still has to solve the linear problem dealing with its potential over- or under-determinedness. For example, practitioners may add regularizations such as the L2-norm of the parameters to ensure that $A^T A$ is nonsingular [16] or add the L1-norm of the parameters to create sparsity [4] in the spirit of minimum-norm solutions to underdetermined problems.

A further complication is that accurately minimizing the energy functional may not even lead to the desired solution since the energy functional is often only suggestive of intent. This is the classical problem of overfitting, *e.g.* minimizing the loss while obtaining large generalization errors on holdout data. In the specific case of facial expression estimation, matching pre-drawn contours may minimize the energy functional at the cost of implausible deformations of the rest of the face. Thus, domain specific regularizers such as Laplacian smoothing of surface meshes [1] or anatomical simulation of the underlying muscle model [9] help to alleviate overfitting. In the case of training neural networks, one can utilize early stopping, dropout, and batch normalization to achieve models that better generalize to more data [13].

To summarize, linear subproblems may not lead to the solution, often require modifications for solvability anyway, and may come from an energy functional which is only at best suggestive of intent. Thus, we take an aggressive approach to solving these linear subproblems by first pruning away any coordinates that are uncorrelated with the right hand side as motivated by least angle regression (LARS) [12]. The other coordinates are then estimated in a coordinate descent fashion [32] eliminating the need to regularize for solvability. Our aim is to find a set of parameters that give a meaningful interpretation of the observed data; thus, using uncorrelated directions to make progress on linear subproblems seems unnecessary if not unwise. The notion that solving linear systems may not necessarily help to solve a nonlinear problem is not new; there are many nonlinear problems where one looks at linearizations only to motivate a strategy for tackling the nonlinear problem –

a quite common example is eigensystem consideration for solving hyperbolic conservation laws, see *e.g.* [35] and the references therein.

We illustrate the efficacy of our approach on a number of problems in the context of facial expression estimation especially when targeting rotoscope curves. The end goal is to determine a set of model parameters that represent the correct semantic meanings including pose and expression while minimizing the excitation of spurious degrees of freedom.

2. Related Work

Regularization: When working with ill-posed problems, regularization is used to prevent the model from over-fitting to the data [13, 14]. Ridge regression [16] and LASSO [34] are two popular methods for regularizing linear least squares problems; they add constraints equivalent to introducing L2 and L1-norm regularization respectively on the parameters of the model [14]. Using L1-norm regularization has the additional benefit of keeping the solution sparse [4, 11]. Additionally, regularization will help ensure that the normal equations used to solve least squares are well-conditioned. However, regularization forces one to balance the trade-off between matching the data and having smaller parameters values.

Coordinate Descent: At each iteration, coordinate descent chooses a single coordinate direction to minimize. This allows the coordinate descent algorithm to avoid null spaces making it an attractive option for use on ill-posed, poorly conditioned problems. The column can be chosen stochastically [28] or deterministically. Popular deterministic methods for choosing the next search direction include cyclic coordinate descent [20], the Gauss-Southwell and Gauss-Southwell-Lipschitz rule [31], and the maximum block improvement (MBI) rule [8]. Generally, either a line search is performed to determine the step size or a fixed step size/learning rate is used. However, the cost of a line search is prohibitive when the function takes a long time to evaluate (*e.g.* in the case of simulations) so we instead use coordinate descent to solve the linear problem found at every iteration of Gauss-Newton. Instead of looking at a single column at a time, block coordinate descent can be used to update multiple columns simultaneously [36]; however, with block coordinate descent, regularization may still be needed to avoid ill-posed problems when the block of columns does not have full rank. A more complete overview of coordinate descent methods can be found in [32, 37].

Correlation: Although coordinate descent algorithms can avoid the null space, they may still choose poorly correlated coordinates potentially resulting in many poorly correlated coordinates in place of fewer more strongly correlated coordinates. Using correlation to choose the next coordinate to add to the model can alleviate this problem and

is the central idea behind MBI [8], forward and backward stepwise regression [10], and least angle regression (LARS) [12]. The latter statistical regression methods are often used to gain better prediction accuracy and interpretability of the model [14]. However, LARS converges to the least squares solution [12] which often means eventually using uncorrelated coordinates.

Faces: When solving for facial blendshape parameters [21], L2-norm regularization of the parameters is commonly used by practitioners to obtain more reasonable solutions due to its ease of use in commodity least squares solvers [2, 5, 22, 33]. Others instead choose to bound the blendshape weights between some minimum and maximum value (typically -1 and 1 or 0 and 1) [1, 17, 23]. The usage of the L1-norm regularization is relatively rare, even though it results in sparser solutions [3, 19, 29]. Other methods use Laplacian regularization [1, 18] or anatomically motivated regularization [38] to constrain the deformation of the face.

3. Motivation

Consider $Ax = b$ where $A = \begin{bmatrix} 1 & -1 \\ .1 & 1 \times 10^{-6} \end{bmatrix}$ and $b = \begin{bmatrix} 0.0 & 1.0 \end{bmatrix}^T$, and the two columns of A get over-dialed by the exact solution of $x_1 = 1/(1 \times 10^{-6} + .1)$ and $x_2 = 1/(1 \times 10^{-6} + .1)$. That is, while b is in fact in the range of A, it is not what we refer to as "easily" in the range of A. Since the columns of A are mostly orthogonal to b, the exact solution shown in Figure 1b overdials both columns producing erroneously large canceling in the horizontal direction in order to edge up vertically even a small amount. Ideally, one would consider these columns of A unrepresentative of b and instead find a control parameter that moves vertically. Practitioners often claim that regularization handles these issues. Regularized least squares of the form $\min_x ||b - Ax||_2^2 + \lambda ||x||_2^2$ has the solution shown in Figure 1c where the non-zero aspect of the regularized solution is almost entirely in the spurious horizontal direc-
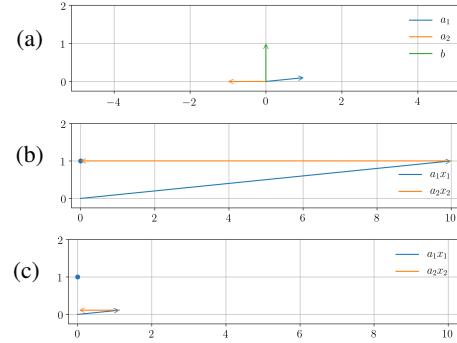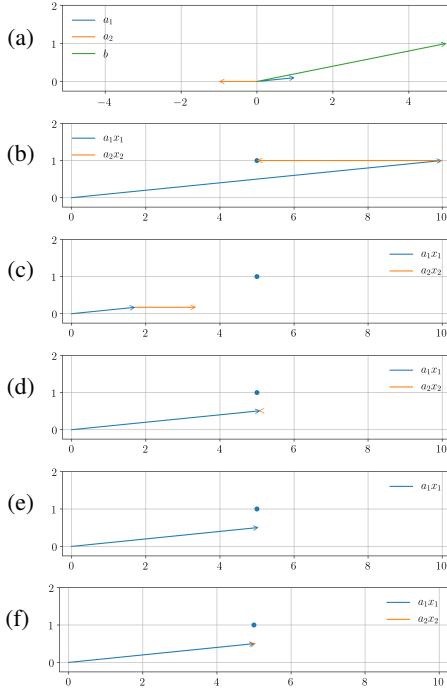

Figure 1. (a) A visualization of the columns of A as well as b for the linear problem $Ax = b$ from Section 3 when $b = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. (b) The exact solution depicted by a_1x_1 and a_2x_2. (c) The regularized solution with $\lambda = .2$.

Figure 2. (a) A visualization of the columns of A as well as b for the linear problem $Ax = b$ from Section 3 when $b = \begin{bmatrix} 5 & 1 \end{bmatrix}^T$. (b) The exact solution depicted by a_1x_1 and a_2x_2. (c) The regularized solution with $\lambda = 1$. (d) The regularized solution with $\lambda_1 = 0$ and $\lambda_2 = 1$. (e) The solution when solving for x_1 only. (f) The solution using coordinate descent using the MBI selection rule after a few iterations.

tion with only an infinitesimal component in the direction of b. In fact, the best one can hope for is to drive the spurious horizontal component to zero with no hope of moving vertically towards the solution.

Next consider the case where $b = \begin{bmatrix} 5 & 1 \end{bmatrix}^T$ and the $b_1 = 5$ component is more readily captured by the columns of A. Here, the exact solution shown in Figure 2b is an improvement over Figure 1b since less of a_1 is wasted to cancel all of a_2, and similarly, the regularized solution shown in Figure 2c makes some progress towards the actual solution as compared to Figure 1c. Still, one could do better by changing the regularization in the least squares problem to have the form $\min_x ||b - Ax||_2^2 + \lambda_1 x_1^2 + \lambda_2 x_2^2$ with $\lambda_1 = 0$. Figure 2d shows the result for $\lambda_2 = 1$ which is highly improved. One could do even better using only a_1 as shown in Figure 2e obtained using $\min_{x_1} ||b - a_1x_1||_2^2$. This is equivalent to minimizing $Ax - b$ using coordinate descent after using only one column.

4. Pruning

We illustrate our approach by solving a generic nonlinear least squares optimization problem of the form $\min_x ||f(x)||_2^2$ using a Gauss-Newton based method, computing the Jacobian $J = \partial f / \partial x$ to utilize the first-order

Taylor expansion $f(x + \delta x) = f(x) + J\delta x$ at each iteration. The minimization problem at the current iteration is $\min_{\delta x} ||f(x) + J\delta x||_2^2$. This is a linear least squares problem and can be solved using $J^T J\delta x = -J^T f(x)$ to find the δx that makes progress towards the solution. The least squares solution can also be computed by solving $J\delta x = -f(x)$ using QR factorization; however, it is not clear that the least squares solution is necessary or desirable.

We thus propose an alternative approach to solving $J\delta x = -f(x)$ which depends on computing the correlation between each column j_i of J and $-f(x)$. Similarly to LARS [12] and MBI [8], we use the dot product magnitude $|\hat{j}_i \cdot f(x)|$ to determine correlation, where $\hat{j}_i = j_i / ||j_i||_2$. We first prune out columns j_i that are poorly correlated with f as these j_i make no progress towards the solution in the linearized state of the problem. Furthermore, these columns can only act as corrections on actual progress in conjunction with other columns; however, these corrections are only valid when the problem is truly linear. In the case of a nonlinear problem, it is unclear whether these uncorrelated corrections help to minimize the nonlinear $||f(x)||_2^2$. See Figure 3. Motivated by the Gauss-Southwell rule, one might instead prune using the dot product computed using $|j_i \cdot f(x)|$ to consider large decreases in residual with small steps; however, this may leave columns that are large and poorly correlated unpruned. Thus, we use the absolute dot product of the normalized vectors as the correlation measure for its geometric interpretation as the absolute cosine of the angle between j_i and $f(x)$.

Pruning columns of J to get a reduced J_S has the additional benefit of potentially eliminating portions of the null space of J, as the pruned out columns may not have full rank or a combination of the pruned out columns with the remaining columns may not have full rank; this pruning may also improve the condition number. This is especially
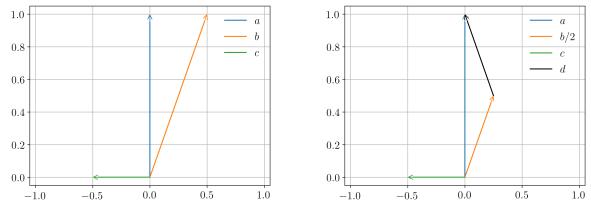

Figure 3. Left: Here, $a = b + c$, but c is only valid for making progress towards a in conjunction with b. While using c may be desirable when trying to solve a linear system of equations, using c (which is perpendicular to a) to progress when solving a linearized high-dimensional nonlinear problem may not produce any meaningful progress. Right: In fact, it may be desirable to only progress in the direction of b for a while, stopping before the remaining residual starts to become uncorrelated. At that point, it would be better to find a new direction (in this case d) that leads back towards a.

prudent when working with a large number of dimensions in which case the dimension of the null space of J and the condition number of J may be large. Moreover, this problem is exacerbated when regularization is not used. At this point, one could compute a modified step δx_S that uses the columns of J_S to make progress towards $-f(x)$. However, if J_S contains a null space, some regularization would still be needed to obtain a reasonable solution.

5. Solving the Pruned System

To avoid regularization, we pursue a coordinate descent strategy to solve $J_S \delta x_S = -f(x)$. At each iteration of the coordinate descent algorithm, a single column j_i of the Jacobian is used to make progress towards $-f(x)$. Since only one column is examined at any given iteration, we do not have to use regularization to avoid null spaces. There are a few strategies for choosing the column to update that follow our motivation of using correlated columns. For example, one could choose the column with the largest correlation with the residual or choose the column which results in the largest decrease in the residual similar to the MBI rule. Motivated by the Gauss-Southwell rule [31], we instead choose the column that maximizes the reduction in residual for a given step, which in effect balances correlation with residual reduction.

Assuming that we measure the size of the residual using the squared L2 norm, we choose the column j_i that maximizes

$$\frac{\Delta(r^T r)}{\Delta \alpha_i} = \frac{||r(\delta x_S)||_2^2 - ||r(\delta x_S) - \alpha_i j_i||_2^2}{|\alpha_i|} \quad (1)$$

where $r(\delta x_S) = -f(x) - J_S \delta x_S$ is the residual for the linearized problem and α_i is the step size when j_i is the coordinate direction. Flipping all j_i so that $r(\delta x_S)^T j_i > 0$ leads to $\alpha_i > 0$, which in turn means that finding the column j_i that maximizes Equation 1 is equivalent to the column j_i that maximizes $M = 2r(\delta x_S)^T j_i - \alpha_i ||j_i||_2^2$.

In the case where α_i is chosen to make as much progress as possible, i.e. $\alpha_i j_i = (r(\delta x_S) \cdot \hat{j}_i)\hat{j}_i$ or $\alpha_i = r(\delta x_S)^T j_i / ||j_i||_2^2$, we obtain $M = r(\delta x_S)^T j_i$ which is standard for Gauss-Southwell. Similarly, as $\alpha_i \to 0$, $M \to 2r(\delta x_S)^T j_i$. Geometrically, this maximizes the projection of j_i into the direction of $r(\delta x_S)$, i.e. $j_i \cdot \hat{r}(\delta x_S)$, since $||r(\delta x_S)||$ does not vary when considering which j_i to use. In comparison, the MBI rule chooses the j_i that maximizes $\hat{j}_i \cdot \hat{r}(\delta x_S)$ ignoring the magnitude of j_i, which unfortunately allows for directionally correlated but small magnitude columns that may require large overdialed α_i to make progress.

When more conservatively choosing α_i less than the greedy value to make the maximal progress, $\alpha_i < r(\delta x_S)^T j_i / ||j_i||_2^2$ and $M = r(\delta x_S)^T j_i + r(\delta x_S)^T j_i - \alpha_i ||j_i||_2^2$. Whereas the greedy choice eliminated the last two
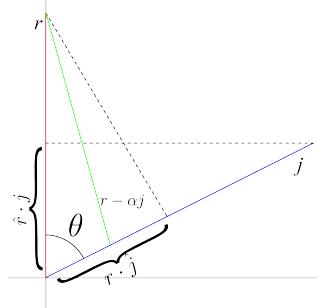

Figure 4. Whereas MBI simply minimizes θ, Gauss-Southwell and our approach instead consider $\hat{r} \cdot j$. However, unlike Gauss-Southwell, our approach also prefers that the remaining residual $r - \alpha j$ still be correlated with j. This penalizes going too far in a search direction j, especially since going too far can lead to using uncorrelated directions as corrections as shown in Figure 3. Instead, we stop progress along j while the remaining residual is still correlated, which allows us to look for more correlated directions to continue making progress similarly to LARS [12].

terms in M giving a result in line with Gauss-Southwell, our approach augments the Gauss-Southwell like term with $r(\delta x_S)^T j_i - \alpha_i ||j_i||_2^2 = (r(\delta x_S) - \alpha_i j_i) \cdot j_i$ which compares the remaining residual to the search direction j_i. Maximizing this latter term helps to prefer search directions that remain correlated with the new residual preventing uncorrelated gains that might later be removed using uncorrelated directions as in Figure 3 (left). See Figure 4. We note that it may be desirable to replace our proposed Heaviside pruning (in Section 4) with a differentiable soft-pruning penalty term in Equation 1.

We generally only execute a few iterations of coordinate descent using these update rules to mimic the regularization effects of early stopping [13] and truncated-Newton methods [27] as it prevents overfitting to the current linear model. Furthermore, we also terminate the coordinate descent iterations early if the decrease in L2 error is low. Stopping early also prevents us from reaching the undesirable least squares solution as in LARS [12].

6. Experiments

Consider matching hand-drawn rotoscope curves on a captured image to the corresponding two-dimensional projections of similar curves drawn on a three-dimensional face model. The face surface $x(w)$ is driven by blendshapes [21] and linear blend skinning for the jaw; let w represent the full set of controls for the jaw angles, jaw translation, and face blendshapes. Assuming the pointwise correspondences between the curves are known, we solve a nonlinear least squares problem of the form

$$\min_w ||C^* - C(x(w))||_2^2 \quad (2)$$

to estimate w, where C^* are the points on the rotoscope curves and $C(x(w))$ are the corresponding points from the
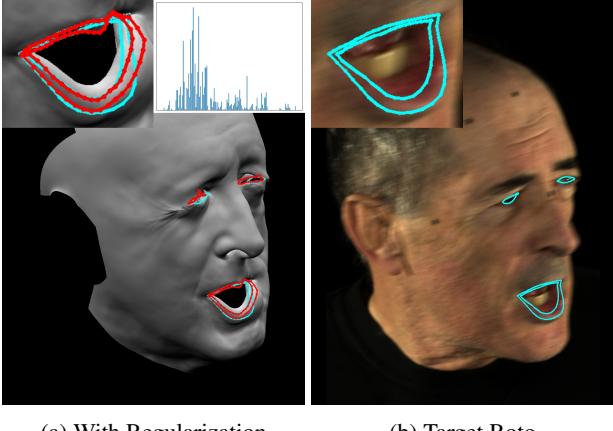
face model projected into the image plane. Here, we assume that the camera parameters and the rigid alignment are precomputed.

To solve this problem, practitioners often add L2 regularization of the parameters (*e.g.* $\lambda \|w\|_2^2$). Although this achieves reasonable results when looking at the geometry and/or synthetic render, the resulting parameter values w are typically densely activated making it nearly impossible to utilize those values for editing or for extracting semantic information. See Figure 5.

6.1. Synthetic Tests

We manually set the two components of w^* (left and right) corresponding to the smile expression to one to produce a face surface $x(w^*)$ from which the barycentrically embedded contour points can be projected into the image plane generating a synthetic target C^*. As a baseline, we then solve Equation 2 using the Dogleg method with no prior, the Dogleg method with a prior weight of $\lambda = 3600$, and the BFGS method [30] with a soft-L1 prior with a weight of 3600 (*i.e.* with an extra term $3600 \sum_i 2(\sqrt{1 + w_i^2} - 1)$ [6]). For our approach, we firstly prune all columns of the Jacobian whose angle to the residual has an absolute cosine less than 0.3 (determined experimentally). Then, α_i is set to a fixed size with $\tau = 1 \times 10^{-2}$ and the coordinate descent algorithm is run until the linear L2 error no longer sufficiently decreases or when more than 10 coordinates have been used. We limit all four methods to take no more than 10 Gauss-Newton iterations and show the results in Figure 6a.

Next, we add an increasing amount of uniformly dis-

tributed noise to the image locations of the targeted contour points. As expected, although Dogleg with no regularization produces reasonable results when the noise is low, as the amount of noise increases, it begins to overfit to the errorneous data. See Figure 6, top row. Both of the regularized approaches as well as our approach, however, are able to target the noisy curves (without overfitting) producing more reasonable geometry. See Figure 6, bottom three rows. Also, as seen in Figure 7, our approach gives more accurate semantics even in the presence of noise.

6.2. Real Image Data

For each captured image, an artist draws eight curves around the mouth and eyes. Another artist hand-selects the corresponding contours on the three-dimensional face

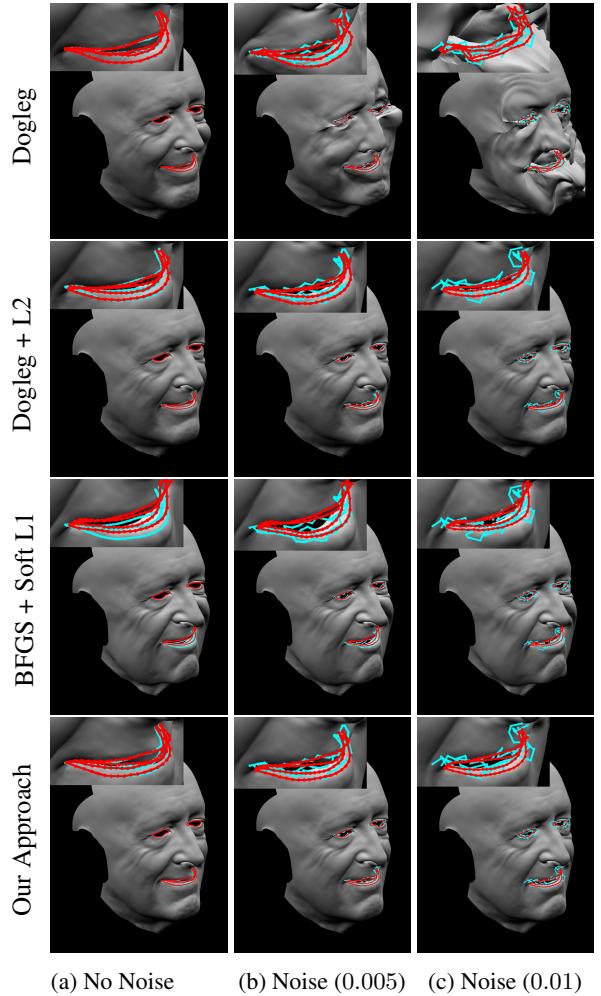(a) No Noise   (b) Noise (0.005)   (c) Noise (0.01)

Figure 6. As we increase the amount of noise added to the points on the target curve, the Dogleg method without regularization overfits causing the mesh to "explode" in spite of having the smallest error as measured by Equation 2 (typical of overfitting). On the other hand, the standard regularization terms and our approach prevent the model from overfitting to the noisy curves.

(a) With Regularization   (b) Target Roto

Figure 5. Using L2 regularization to solve the minimization problem in Equation 2. The teal lines show the target curves and the red lines show the corresponding contours from the face model projected into the image plane. The inset bar graph shows the values for the 146 control parameters. While the resulting geometry looks reasonable, the resulting parameter values are semantically meaningless and unrepresentative of the actual facial expression.

(a) MBI   (b) GS   (c) Ours

Figure 8. We compare the coordinates chosen by the MBI rule, the Gauss-Southwell (GS) rule, and our approach. The top row are the results after a single Gauss-Newton iteration, and the bottom row are the results after 10 Gauss-Newton iterations.
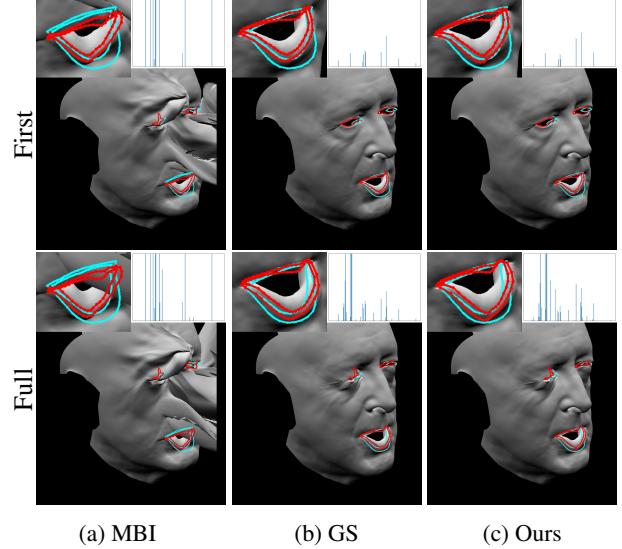
(a) No Noise   (b) Noise (0.005)   (c) Noise (0.01)

Figure 7. The corresponding blendshape weights from Figure 6. The target solution was generated by setting the two orange columns to one and the blue columns to zero. The figure heights are clipped at 1.0 and many parameter values exceed that. Though the reguarized solves have smaller, spurious weights than the non-regularized version (second and third row vs. first row), our approach (last row) produces a much sparser solution with more semantic meaning even in the presence of noise.

model which are projected into the image plane using camera extrinsic and intrinsic parameters calibrated using the method of [15]. Generally, these two sets of contours will contain a different number of control points; as a result, we uniformly resample both in the image plane. The uniformly sampled curves are then used in Equation 2.

Our approach and using regularization give the most reasonable geometric results – similar to the synthetic case. However, in certain frames our approach seems to over-regularize the result and cause the mouth to not quite hit the desired expression. This could be due, in part, to not having a fully accurate rigid alignment. However, a major benefit of our method is the resulting sparsity in the weights. While the Dogleg method with and without regularization will dial in nearly all the parameters to a non-zero value, our approach generally dials in a small number of weights. See Figure 11. Similarly, the soft L1 regularized solution also produces sparser solutions than the L2 regularized solution; however, due to approximations in the chosen optimization approach (BFGS, Soft L1), it produces many small (*i.e.* $< 1 \times 10^{-3}$) weights instead of zero values. While one could use a threshold to clamp values to zero, care must be taken to not accidentally threshold a small blendshape weight (or a combination of small blendshape weights) that contribute significantly to the overall performance. We note

that one could potentially use iteratively reweighted least squares methods [7] to obtain sparse solutions.

6.3. Parameter Study

Column Choice: We compare Equation 1 for choosing the coordinate descent column to using Gauss-Southwell and MBI without pruning. Firstly, for each approach, we linearize and solve with no thresholding for the relative decrease in L2 error, an upper limit of 10 unique coordinates used, and a fixed step size of 0.01. Figure 8 (top row) shows similar behavior in terms of geometry and choice of blend-shape weights between our approach and Gauss-Southwell. The MBI rule, on the other hand, immediately overfits and dials in extraordinarily high weights, especially for blend-shapes that affect the eyes, causing the face surface to explode; however, as can be seen in the supplementary material, solving without the eye rotoscope curves causes MBI to overfit to the mouth curves instead. Secondly, repeating the linearization and solve 10 times as shown in Figure 8 (bottom row) gives similar results.

Step Size: Recall α_i are the parameters, *i.e.* blend-shape weights, that generally vary between 0 and 1; thus we choose to compare fixed step sizes of $\tau = 0.01, 0.02, 0.1, 0.5$ and 1.0 to the full, greedy step, *i.e.* $\alpha_i = r^T j_i / \|j_i\|_2^2$. Again without pruning, we run 10 Gauss-Newton iterations with no thresholding for the relative decrease in L2 error and an upper limit of 10 unique coordinates used. As seen in Figure 9, the smaller step sizes achieve better overall facial shapes and less overdialed weights. In particular, the greedy step dials 7 weights to be greater than 1 while the $\tau = 0.01$ and $\tau = 0.02$ step sizes only dial in 4 such weights. Removing the eye ro-
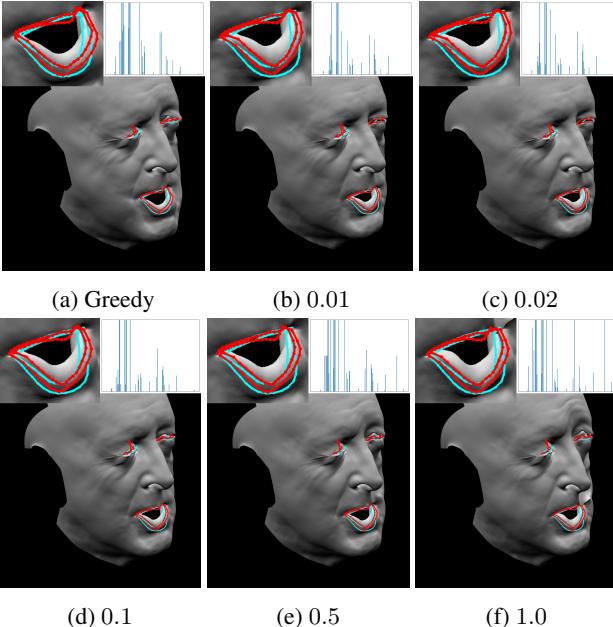
(a) Greedy　　　　(b) 0.01　　　　(c) 0.02

(d) 0.1　　　　(e) 0.5　　　　(f) 1.0

Figure 9. We compare the behavior of the geometry and the blendshape weights when using different step sizes.

(a) No Pruning　　　　(b) 0.05　　　　(c) 0.2

(d) 0.3　　　　(e) 0.4　　　　(f) 0.5

Figure 10. As we increase the minimum absolute dot product threshold for pruning, the resulting solution becomes sparser and more regularized.

toscope curves causes the overdialed weights to disappear; however, as seen in the supplementary material, the greedy step causes the mouth to move unnaturally.

We also compare the effect of using fixed step sizes in Equation 1 versus the full, greedy step size equivalent to Gauss-Southwell without pruning. To isolate this variable, we run 10 Gauss-Newton iterations with no thresholding for the relative decrease in L2 error and an upper limit of 10 unique coordinates used. We vary α_i in Equation 1 but set the actual step size taken to be fixed at 0.01. As shown in supplementary material, while the resulting geometry and weights are all similar, our approach of allowing the step size to influence the chosen coordinate allows the optimization to more quickly reduce the error in earlier Gauss-Newton iterations.

Pruning: We use the correlation metric of $\hat{r}^T \hat{j}_i$ from LARS and MBI to prune poorly correlated coordinate directions. We choose threshold values of 0.0 (no pruning), 0.01, 0.05, 0.1, 0.2, and 0.3, and run 10 Gauss-Newton iterations with a step size of 0.01 with no thresholding for the relative decrease in L2 error. To emphasize the effect of pruning, we allow up to 50 unique coordinates per linearization, and focus only on the rotoscope curves around the mouth. With little to no pruning the model overfits and the geometry around the mouth deforms unreasonably. As the pruning threshold increases, the geometry becomes more regularized and the blendshape weights are less overdialed and sparser as the optimization is forced to use only the most correlated directions. See Figure 10. However, we caution that too much pruning causes MBI style column choices.
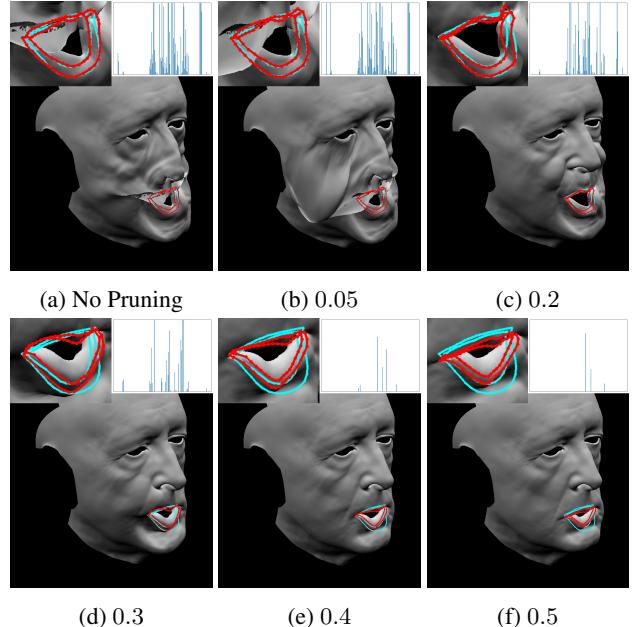
7. Conclusion

Taking the linearization of the nonlinear problem as a mere suggestion for a search strategy, we prune uncorrelated directions, pursue a coordinate descent approach that does not need regularization, and choose search directions in order to maximize gains in reducing the residual with minimal parameter value increases. We stress that this is more generally an outline of an improved search strategy as opposed to a formal and precise new method; yet, our first attempts at such an approach led to highly improved results. In the case of estimating three-dimensional facial expression from a mere eight contours drawn on a single two-dimensional RGB image, we were able to robustly estimate clean, sparse parameter values with good semantic meaning in a highly underconstrained situation where one would typically need significant regularization. In fact, the standard approach without regularization was widly inaccurate, and while regularization helped to moderate the overall face shape, it excited almost every parameter in the model clouding semantic interpretation.

As future work, we would be interested in considering similar search strategies for training neural networks with limited data. Such situations are similar to our example where we only use sparse rotoscope curves that are only suggestive of the desired expression, instead of dense data for every triangle (such as when using shape-from-shading or optical flow).

(a) Dogleg   (b) Dogleg + L2   (c) BFGS + Soft L1   (d) Our Approach   (e) Target
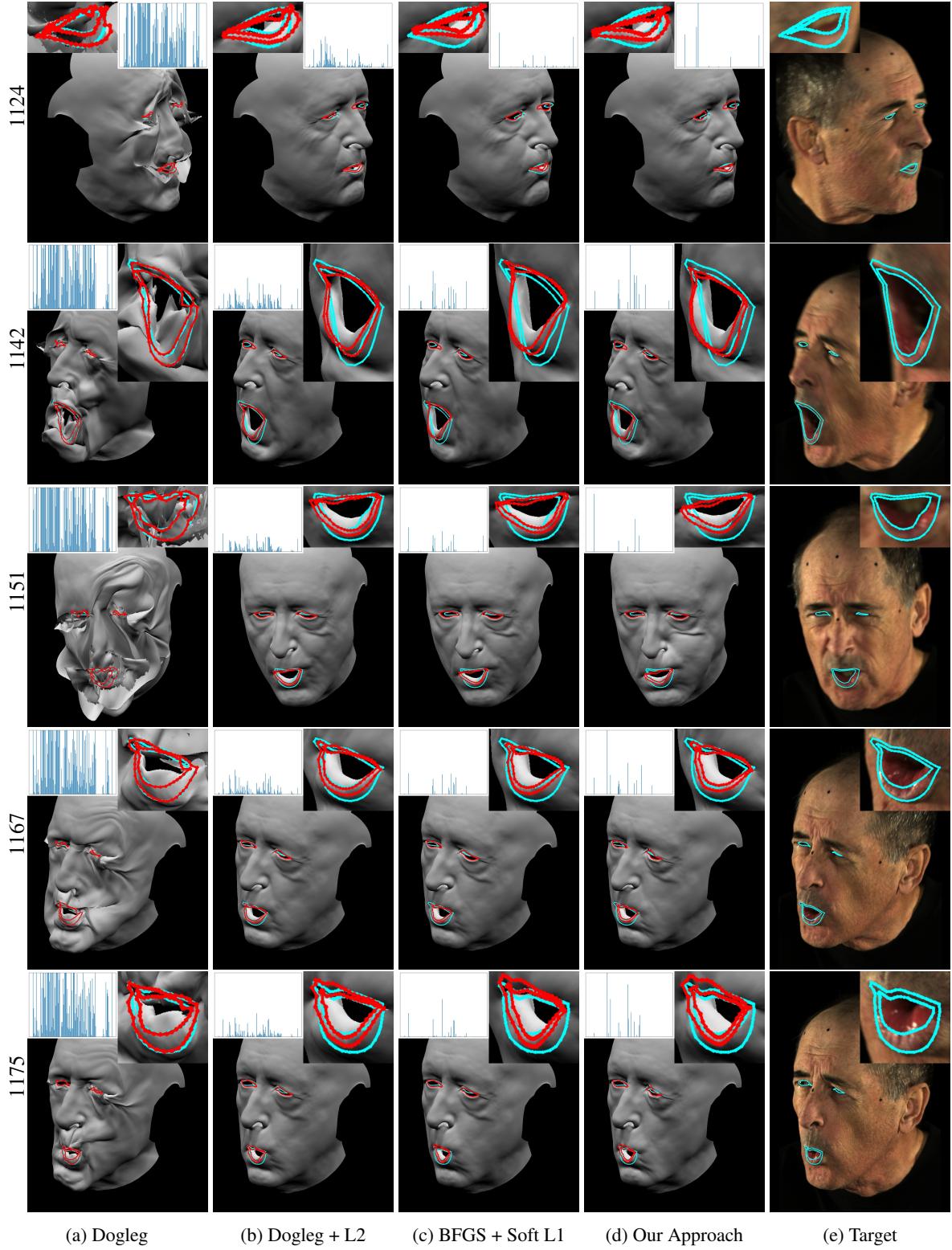
Figure 11. Dogleg without regularization clearly overfits to the curves, producing highly unrealistic face shapes. Dogleg with regularization performs better but sometimes overfits as well. This could be tuned by increasing the regularization weight at the cost of potentially damping out the performance. Our approach produces facial expressions that are reasonably representative of the captured image. The inset bar plots demonstrate the sparsity of the weights for each of the methods. Our method generally produces the sparsest set of weights; *e.g.* on frame 1142, our method has 12 non-zero parameter values while L2 regularization produces fully dense results and soft L1 regularization has 49 significant parameter values (*i.e.* $> 1 \times 10^{-3}$).

Acknowledgements

A. Extended Parameter Study

A.1. Column Choice

Once again, we compare our approach for choosing the next coordinate descent column to using Gauss-Southwell and MBI. For each approach, we linearize and solve with no thresholding for the relative decrease in L2 error, an upper limit of 10 unique coordinates used, and a fixed step size of 0.01; however, in these examples, we remove the eye rotoscope curves from the energy function. Even without solving for the eye rotoscope curves, MBI will still overfit and overdial mouth blendshapes (*e.g.* the two most dialed in shapes have magnitudes of 85.78 and 63.12). On the other hand, Gauss-Southwell and our approach maintain the face surface geometry and keep the blendshape weights within a reasonable range while maintaining the sparsity of the solution. We note that with coordinate descent it is generally a matter of when, not if, the algorithm chooses a coordinate that will be overdialed; our examples demonstrate that MBI reaches those coordinates more quickly than Gauss-Southwell and our approach. See Figure 12.

A.2. Step Size

Once again, we compare fixed step sizes of $\tau = 0.01, 0.02, 0.1, 0.5$ and 1.0 to the full, greedy step, *i.e.* $\alpha_i = r^T j_i / \|j_i\|_2^2$. We again note that the fixed step sizes for each coordinate are clamped to not exceed the full, greedy step size. Again without pruning, we run 10 Gauss-Newton iterations with no thresholding for the relative decrease in L2 error and an upper limit of 10 unique coordinates used; however, in these examples, we remove the eye rotoscope curves from the energy function. In this case, the step size has no significant effect on the sparsity of the blendshape weights; however, as seen in Figure 13, taking the greedy step versus taking a fixed size step results in a more unnaturally shaped mouth. This would seem to indicate that always taking the greedy step will result in some overfitting.

We also run tests with and without the eye rotoscope curves to isolate the effect the step size has on the column choice in our approach; while we vary the step size used to determine the next coordinate descent column, we fix the actual step size to have a magnitude of 0.01. Again without pruning, we run 10 Gauss-Newton iterations with no thresholding for the relative decrease in L2 error and an upper limit of 10 unique coordinates used. We note that when the full, greedy step size is used to choose the next coordinate direction, our approach produces the same results, *i.e.* chooses the same coordinates and takes the same steps, as Gauss-Southwell. As can be seen in Figure 15, the final geometry and blendshape weights when varying the step size are similar to the results produced by Gauss-Southwell, *i.e.* using the full, greedy step. However, our approach, as seen in Figure 14 allows the error to decrease more in earlier Gauss-Newton iterations than when using Gauss-Southwell. Therefore, it may be beneficial to use our approach with a fixed size step when only a few Gauss-Newton iterations are desired.

B. Parameter Limits and Trust Region

Using a coordinate descent solver for the linear problem at every Gauss-Newton iteration allows us to build the solution to $Ax = b$ incrementally without relying on a black box linear solver. Any approach that aims to prevent the solution from going past a certain point (*e.g.* parameter limits and a trust region method) are thus trivial to implement. Given min and max parameters $x_i^{\min}$ and $x_i^{\max}$, at every iteration of the coordinate descent solve, we can clamp each α_i such that $x_i^{\min} \leq x_i + \delta x_i + \alpha_i \leq x_i^{\max}$. Additionally, given a trust region radius μ, we can simply terminate the coordinate descent solve when/if $\|\delta x_S\|_2 > \mu$; this is similar in spirit to how the trust region parameter is used in the Dogleg method. Furthermore, we can introduce limits on α_i to ensure that each step will not cause the solver to go beyond the trust region.

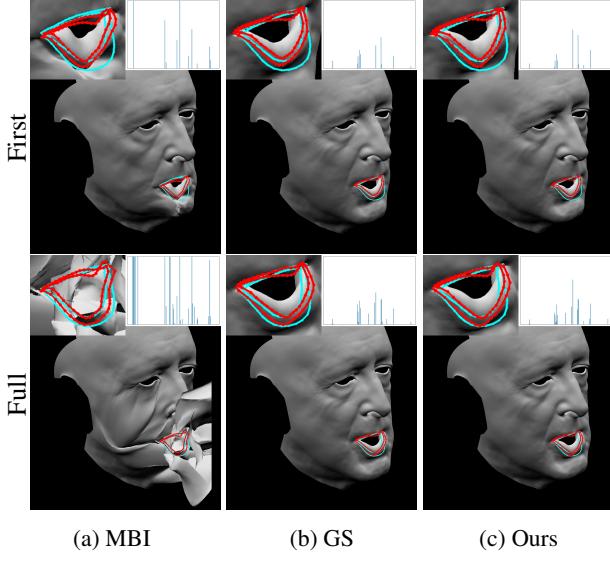(a) MBI   (b) GS   (c) Ours

Figure 12. A comparison of the coordinates chosen by the MBI rule, the Gauss-Southwell (GS) rule, and our approach when solving without the eye rotoscope curves. The top row are the results after a single Gauss-Newton iteration, and the bottom row are the results after 10 Gauss-Newton iterations.

(a) Greedy   (b) 0.01   (c) 0.02

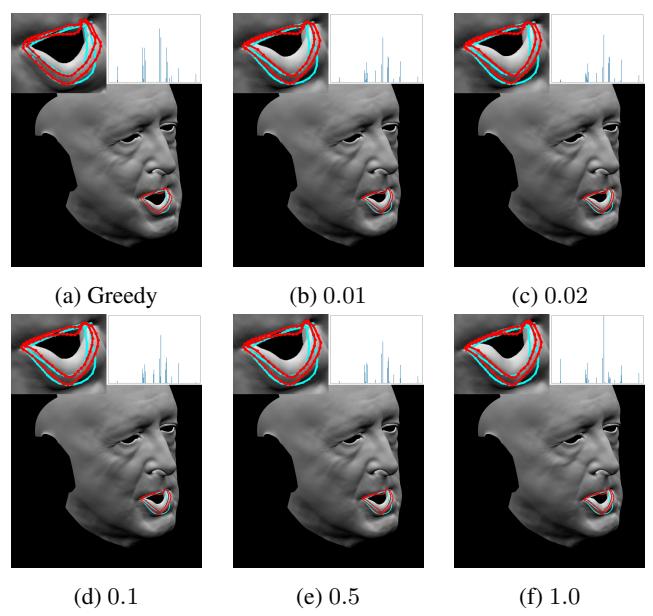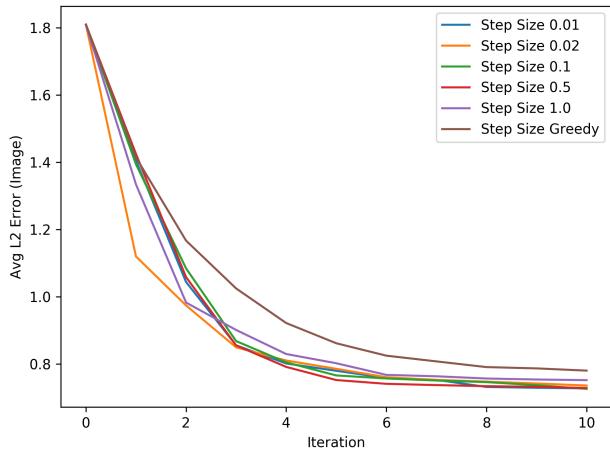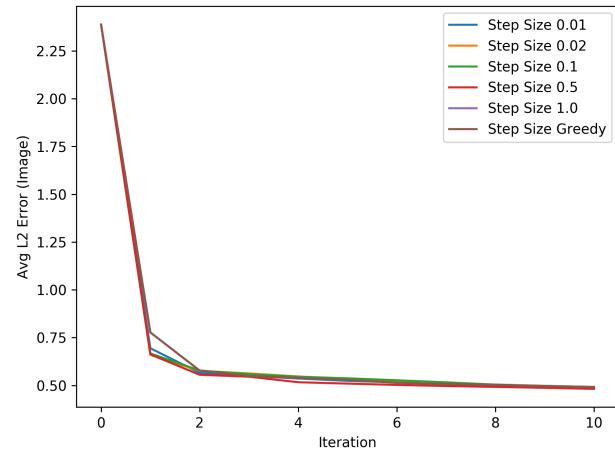(d) 0.1   (e) 0.5   (f) 1.0

Figure 13. A comparison of the behavior of the geometry and the blendshape weights when using different step sizes when solving without the eye rotoscope curves.

(a) with eye rotoscope   (b) no eye rotoscope

Figure 14. A comparison of the average L2 errors plotted before every Gauss-Newton iteration when varying the step size parameter used to choose the next coordinate direction in our approach. The brown lines plot the average L2 errors when using the Gauss-Southwell approach; notice how our approach allows for a faster reduction in error.

with eye rotoscope no eye rotoscope with eye rotoscope no eye rotoscope

Figure 15. A comparison of the geometry and blendshape results from varying the step size parameter used for choosing the next coordinate in our approach.

References

[1] K. S. Bhat, R. Goldenthal, Y. Ye, R. Mallet, and M. Koperwas. High fidelity facial animation capture and retargeting with contours. In *Proceedings of the 12th ACM SIGGRAPH/eurographics symposium on computer animation*, pages 7–14. ACM, 2013.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[3] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):40, 2013.

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014.

[6] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on image processing*, 6(2):298–311, 1997.

[7] R. Chartrand and W. Yin. Iterative reweighted algorithms for compressive sensing. Technical report, 2008.

[8] B. Chen, S. He, Z. Li, and S. Zhang. Maximum block improvement and polynomial optimization. *SIAM Journal on Optimization*, 22(1):87–107, 2012.

[9] M. Cong, K. S. Bhat, and R. Fedkiw. Art-directed muscle simulation for high-end facial animation. In *Symposium on Computer Animation*, pages 119–127, 2016.

[10] S. Derksen and H. J. Keselman. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology*, 45(2):265–282, 1992.

[11] D. L. Donoho. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(6):797–829, 2006.

[12] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[13] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer series in statistics New York, NY, USA:, 2001.

[15] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

[16] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[17] P.-L. Hsieh, C. Ma, J. Yu, and H. Li. Unconstrained realtime facial performance capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1675–1683, 2015.

[18] H. Huang, J. Chai, X. Tong, and H.-T. Wu. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. In *ACM Transactions on Graphics (TOG)*, volume 30, page 74. ACM, 2011.

[19] A. E. Ichim, S. Bouaziz, and M. Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)*, 34(4):45, 2015.

[20] I. Krylov and F. L. Chernous' ko. Solution of problems of optimal control by the method of local variations. *USSR Computational Mathematics and Mathematical Physics*, 6(2):12–31, 1966.

[21] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. H. Pighin, and Z. Deng. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 1(8), 2014.

[22] H. Li, T. Weise, and M. Pauly. Example-based facial rigging. *Acm transactions on graphics (tog)*, 29(4):32, 2010.

[23] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42–1, 2013.

[24] M. Lourakis and A. A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1526–1531. IEEE, 2005.

[25] K. Madsen, H. B. Nielsen, and O. Tingleff. Methods for nonlinear least squares problems. *Informatics and Mathematical Modelling, Technical University of Denmark, DTU*, 2004.

[26] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[27] S. G. Nash. A survey of truncated-newton methods. *Journal of computational and applied mathematics*, 124(1-2):45–59, 2000.

[28] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[29] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics (TOG)*, 32(6):179, 2013.

[30] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

[31] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641, 2015.

[32] H.-J. M. Shi, S. Tu, Y. Xu, and W. Yin. A primer on coordinate descent algorithms. *arXiv preprint arXiv:1610.00040*, 2016.

[33] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[34] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[35] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction.* Springer Science & Business Media, 2013.

[36] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[37] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[38] C. Wu, D. Bradley, M. Gross, and T. Beeler. An anatomically-constrained local deformation model for monocular face capture. *ACM Transactions on Graphics (TOG)*, 35(4):115, 2016.