Global Supervised Descent Method

Xuehan Xiong and Fernando De la Torre
Carnegie Mellon University, Pittsburgh PA
{xxiong,ftorre}@andrew.cmu.edu

Abstract

Mathematical optimization plays a fundamental role in solving many problems in computer vision (e.g., camera calibration, image alignment, structure from motion). It is generally accepted that second order descent methods are the most robust, fast, and reliable approaches for nonlinear optimization of a general smooth function. However, in the context of computer vision, second order descent methods have two main drawbacks: 1) the function might not be analytically differentiable and numerical approximations are impractical, and 2) the Hessian may be large and not positive definite. Recently, Supervised Descent Method (SDM), a method that learns the "weighted averaged gradients" in a supervised manner has been proposed to solve these issues. However, SDM is a local algorithm and it is likely to average conflicting gradient directions. This paper proposes Global SDM (GSDM), an extension of SDM that divides the search space into regions of similar gradient directions. GSDM provides a better and more efficient strategy to minimize non-linear least squares functions in computer vision problems. We illustrate the effectiveness of GSDM in two problems: non-rigid image alignment and extrinsic camera calibration.

1. Introduction

Many computer vision problems (e.g., camera calibration, image alignment, structure from motion) are solved with nonlinear optimization methods. In general, most computer vision-related optimization problems of interest have multiple local minima and are NP-hard to solve. Global optimization algorithms are typically very computationally expensive, have poor convergence properties, and generally suitable for low dimensional search spaces. As a compromise, local optimization methods are usually employed to find a local minimum. Whether global optimization can be solved in polynomial time is still unknown. However, there is a large number of existing techniques that approximate the solution. These techniques include Simulated Annealing [15, 31], Evolutionary algorithms [18], Monte Carlo methods [29, 30] or Branch and Bound [7, 22]. Additionally, Sminchisescu *et al.* [39] proposed an interesting algorithm to systematically traverse nearby local minima by locating the transition states.

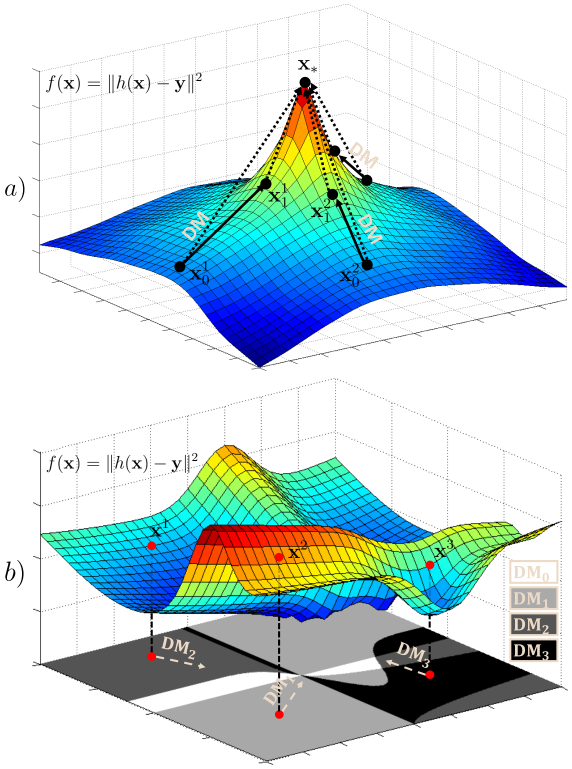


Figure 1. a) A single Descent Map (DM) is used in SDM for minimizing a simple function. b) An example of a more complex objective function. In order to use SDM, its domain has to be split into four regions (represented by different grays) and a separate DM is learned for each region.

Most global optimization methods are computationally expensive and do not guarantee the global optima in polynomial time. On the other hand, local optimization methods based on gradient methods have achieved tremendous success in computer vision problems. When applying gradient-based methods to global optimization, multiple random starts are typically required. Generally, initial values that are close to each other give descent paths that tend to the same minimum point. This phenomenon is formally known as *basin of attraction*, the set of initial values leading to the same local minimum. In the context of local optimization, it is generally accepted that for a general smooth function, second order descent methods are the most robust, fast, and reliable approaches for nonlinear optimization. However, in the context of computer vision, second order descent methods have two main drawbacks: 1) the function might not be analytically differentiable and numerical approximations are impractical, and 2) the Hessian may be large and not positive definite. To address these issues, Xiong and De la Torre [44] proposed a Supervised Descent Method (SDM) for optimizing Nonlinear Least Squares (NLS) functions. Unlike previous methods, it uses supervised data to drive the optimization search. SDM has shown promising results in face alignment [1, 25, 43, 46, 48] and been extended to other computer vision applications such as, object pose estimation [44], rigid object tracking [44], object relocalization [26], human pose estimation [45], and object part localization [45].

Fig. 1a illustrates the idea of SDM. During training, in each iteration SDM learns a single generic Descent Map (DM) from the optimal optimization trajectories (indicated by the dotted lines). In testing, the same DM is used for driving an unseen sample to $\mathbf{x}_*$ (the labeled ground-truth). DM exists under two mild conditions (see Section 2.1). For simple functions, such conditions normally hold. However, in many real applications the function might have several local minima in a relatively small neighborhood, for instance see Fig. 1b for an example. Standard SDM would average conflicting gradient directions resulting in undesirable performance. To overcome this issue, GSDM learns not one but a set of generic DMs (in this example, four), one for different domains (colored by different intensity of grays) of the objective function. Each domain contains only similar gradient directions and one separate DM is learned for each. At iteration k, $\mathbf{x}_k$ may step into any of the four regions and the corresponding DM is used to update the result. Based on this intuition, this paper introduces and validates a new concept, domain of homogeneous descent and extends the theory of SDM to global optimization. In addition, we discover the connection between SDM and Imitation Learning and develop a practical algorithm based on GSDM to track faces from profile to profile and illustrate how GSDM can also be used for extrinsic camera calibration.

2. Theory

In this section, we extend the theory of SDM to deal with multiple local minima. First, we review the concept of DM and the two conditions for it to exist. Second, we discuss SDM's interesting connection with Imitation Learning. Then, we introduce and validate a new concept termed, *domains of homogeneous descent*.

2.1. Review of SDM

This section reviews SDM originally introduced in [44] and its theoretical properties. See footnote[1] for notation. Given a Nonlinear Least Squares (NLS) problem,

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} \|\mathbf{h}(\mathbf{x}) - \mathbf{y}\|^2, \tag{1}$$

where $\mathbf{h}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ is a nonlinear function, $\mathbf{y} \in \mathbb{R}^m$ is a known vector, and $\mathbf{x} \in \mathbb{R}^n$ is the optimizing parameter. Applying the chain rule to Eq. 1, the gradient descent update rule yields

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha \mathbf{A} \mathbf{J}_{\mathbf{h}}^{\top}(\mathbf{x}_{k-1})(\mathbf{h}(\mathbf{x}_{k-1}) - \mathbf{y}) \tag{2}$$

where $\mathbf{J}_{\mathbf{h}}(\mathbf{x}) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the identity ($\mathbf{I}_n$) in first-order methods, or the inverse Hessian (or an approximation) for second-order methods, and α is the step size. Computing the rescaling factor $\mathbf{A}$ and gradient direction $\mathbf{J}_{\mathbf{h}}$ in high-dimensional spaces is computationally expensive and can be numerically unstable, especially in the case of non-differentiable functions, where finite differences are required to compute estimates of the Hessian and Jacobian. The main idea behind SDM is to avoid explicit computation of the Hessian and Jacobian and learn the generic DM ($\mathbf{R} \sim \alpha \mathbf{A} \mathbf{J}_{\mathbf{h}}(\mathbf{x}_{k-1})$) from training data. Note that DM is not a descent direction, it contains part of descent direction and needs to multiply with $(\mathbf{h}(\mathbf{x}) - \mathbf{y})$ to produce a descent direction. Alternatively, $\mathbf{R}$ can be seen as the "weighted average gradient direction" of $\mathbf{h}$ around $\mathbf{x}_*$. We define $\mathbf{R}$ more formally below.

Definition 1. *A matrix* $\mathbf{R} \in \mathbb{R}^{n \times m}$ *is called a* generic descent map *if there exists a scalar* $0 < c < 1$ *such that* $\forall \mathbf{x} \in N(\mathbf{x}_*)$, $\|\mathbf{x}_* - \mathbf{x}_k\| \leq c\|\mathbf{x}_* - \mathbf{x}_{k-1}\|$. $\mathbf{x}_k$ *is updated using the following equation:*

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{R}(\mathbf{h}(\mathbf{x}_{k-1}) - \mathbf{h}(\mathbf{x}_*)). \tag{3}$$

Xiong and De la Torre [44] proved the existence of a generic DM under the following conditions: 1). $\mathbf{R}\mathbf{h}(\mathbf{x})$ is a strictly locally monotone operator anchored at the optimal solution $\mathbf{x}_*$. 2). $\mathbf{h}(\mathbf{x})$ is locally Lipschitz continuous anchored at $\mathbf{x}_*$.

2.2. SDM as Policy Learning

Imitation Learning (IL) can be seen as a special case of Supervised Learning. In Supervised Learning, the agent is

[1] Bold capital letters denote a matrix $\mathbf{X}$; bold lower-case letters denote a column vector $\mathbf{x}$. All non-bold letters represent scalars. $\mathbf{x}_i$ represents the ith column of the matrix $\mathbf{X}$. x_{ij} denotes the scalar in the i^{th} row and j^{th} column of the matrix $\mathbf{X}$. x_j denotes the scalar in the j^{th} element of $\mathbf{x}$. $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is an identity matrix. $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}$ denotes the Euclidean distance. $\|\mathbf{X}\|_F = \sqrt{tr(\mathbf{X}^T\mathbf{X})} = \sqrt{tr(\mathbf{X}\mathbf{X}^T)}$ designates the Frobenious norm of a matrix. $N(\mathbf{x})$ denotes the neighborhood of $\mathbf{x}$.

presented with labeled training data and learns an approximation to the function that produced the data. Within IL, this training dataset is composed of example executions (sequences of state and action pairs) of the task by a demonstration teacher. The goal is to derive a policy that reproduces the demonstrated behavior. The world consists of states S, actions A, and a policy is the mapping between the two. In real-world applications, the state is often not fully observable and the learner instead has access to an observed state Z.

In the context of minimizing a NLS function, $\mathbf{y}$ is regarded as the desired state and the objective is to find the action $\Delta\mathbf{x}$ that moves from the initial state to the desired state. The nonlinear function $\mathbf{h}$ is the feature function that partially represents the state. The demonstration data contains a set of observation and action pairs. The observed states are represented by a set $Z = \{\mathbf{h}(\mathbf{x}^i) - \mathbf{y}^i\}$ of errors (misalignments) between the known vectors $\{\mathbf{y}^i\}$ and the function evaluations at the current parameter estimates $\{\mathbf{h}(\mathbf{x}^i)\}$. The action set will correspond to the parameter updates $A = \{\Delta\mathbf{x}^i\}$, and the policy maps misalignments to parameter updates. In SDM, the policy is derived as a sequence of linear mapping functions between states and actions. Within this context, the teacher is always available for giving feedback. More specifically, since the ground truth solutions $\{\mathbf{x}_*^i\}$ are available throughout training, the teacher can always give the perfect action based on the state observation. SDM takes advantage of this fact by learning not one but a sequence of policies so the latter ones correct mistakes made from previous iterations after the teacher's feedbacks.

2.3. Domains of Homogeneous Descent

For a function f with a unique minimum, the gradients of $\mathbf{h}$ often share similar directions. Therefore, a weighted average can be learned. When dealing with the function f with several local minima (See Fig. 1b for an example), the gradients of $\mathbf{h}$ may have conflicting directions so averaging them is not adequate and it may cause the SDM training to stall. Later in the paper, we validate this intuition in our experiments on extrinsic camera calibration. When we enlarge the parameter space, the performance of SDM drops dramatically.

We will bypass this problem by learning **not one** but **a set** of DMs. In the following, we will prove that it is possible to find a partition of domain $\mathbf{x}$, $S = \{S^t\}_1^T$, such that there exists a generic DM $\mathbf{R}^t, \forall t, \mathbf{x} \in S^t$. The subsets of this partition is defined as *Domains of Homogeneous Descent (DHD)*. In Fig. 2 we plot four NLS functions along with their DHD found by following the strategy proposed in Theorem 1. Interestingly, local minima are located at the intersections of different domains.

Theorem 1. *Under one minor condition:*

$$\exists K > 0, \frac{\|\mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{x}_*)\|}{\|\mathbf{x} - \mathbf{x}_*\|} < K, \forall \mathbf{x} \in N(\mathbf{x}_*), \tag{4}$$

there exists a finite partition of domain $\mathbf{x}$, $S = \{S^t\}_1^T$, *such that* $\forall \mathbf{x} \in S^t$, *there exists a generic DM* $\mathbf{R}^t$.

Proof. To simplify the notation, we denote $\mathbf{x}_* - \mathbf{x}$ as $\Delta\mathbf{x}$ and $\mathbf{h}(\mathbf{x}_*) - \mathbf{h}(\mathbf{x})$ as $\Delta\mathbf{h}$. We will prove the above theorem by finding a specific partition with its corresponding DMs. Let us consider a partition strategy based on the signs of $\Delta x_j \Delta h_j$. Each sign can take on two values ± 1 and j ranges from 1 to $\min(n, m)$. Each subset of this partition contains $\mathbf{x}$ that satisfy one of the $2^{\min(n,m)}$ unique conditions. Without loss of generality, let us derive the DM for the subset S^0 where $\forall j, sign(\Delta x_j \Delta h_j) = 1$. We want to show that there exists a $\mathbf{R}$ such that

$$\frac{\|\mathbf{x}_* - \mathbf{x}_k^i\|}{\|\mathbf{x}_* - \mathbf{x}_{k-1}^i\|} < 1, \text{if } \mathbf{x}_* \neq \mathbf{x}_{k-1}^i. \tag{5}$$

We replace $\mathbf{x}_k^i$ with $\mathbf{x}_{k-1}^i$ using Eq. 3 and squaring the left side of Eq. 5 gives us,

$$\begin{aligned}
&\frac{\|\Delta\mathbf{x}_k^i\|^2}{\|\Delta\mathbf{x}_{k-1}^i\|^2} = \frac{\|\Delta\mathbf{x}_{k-1}^i - \mathbf{R}\Delta\mathbf{h}_{k-1}^i\|^2}{\|\Delta\mathbf{x}_{k-1}^i\|^2} \\
=&\frac{\|\Delta\mathbf{x}_{k-1}^i\|^2}{\|\Delta\mathbf{x}_{k-1}^i\|^2} + \frac{\|\mathbf{R}\Delta\mathbf{h}_{k-1}^i\|^2}{\|\Delta\mathbf{x}_{k-1}^i\|^2} - 2\frac{\Delta\mathbf{x}_{k-1}^{i\top}\mathbf{R}\Delta\mathbf{h}_{k-1}^i}{\|\Delta\mathbf{x}_{k-1}^i\|^2} \\
=&1 + \frac{\|\mathbf{R}\Delta\mathbf{h}_{k-1}^i\|}{\|\Delta\mathbf{x}_{k-1}^i\|^2}\left(\|\mathbf{R}\Delta\mathbf{h}_{k-1}^i\| - 2\Delta\mathbf{x}_{k-1}^{i\top}\frac{\mathbf{R}\Delta\mathbf{h}_{k-1}^i}{\|\mathbf{R}\Delta\mathbf{h}_{k-1}^i\|}\right).
\end{aligned} \tag{6}$$

Setting Eq. 6 < 1 gives us,

$$\|\mathbf{R}\Delta\mathbf{h}_{k-1}^i\| \leq 2\Delta\mathbf{x}_{k-1}^{i\top}\frac{\mathbf{R}\Delta\mathbf{h}_{k-1}^i}{\|\mathbf{R}\Delta\mathbf{h}_{k-1}^i\|}. \tag{7}$$

The choice of $\mathbf{R}$ needs to guarantee that the right side of Eq. 7 is greater than zero. Remember that in subset $S^{(0)}$ $sign(\Delta x_j \Delta h_j) = 1, \forall j$. A trivial $\mathbf{R}$ would be $c\mathbf{D}$, where $c > 0$ and $\mathbf{D}$ is a rectangular diagonal matrix with the diagonal elements equal to 1. From the geometric definition of dot product, we can rewrite the right side of the inequality 7 as,

$$2\Delta\mathbf{x}_{k-1}^{i\top}\frac{\mathbf{R}\Delta\mathbf{h}_{k-1}^i}{\|\mathbf{R}\Delta\mathbf{h}_{k-1}^i\|} = 2\|\Delta\mathbf{x}_{k-1}^i\|\cos\theta^i,$$

where θ^i is the angle between vectors $\Delta\mathbf{x}_{k-1}^i$ and $\mathbf{R}\Delta\mathbf{h}_{k-1}^i$. Using the condition stated in 4 we have

$$2\|\Delta\mathbf{x}_{k-1}^i\|\cos\theta^i \geq \frac{2}{K}\|\Delta\mathbf{h}_{k-1}^i\|\cos\theta^i. \tag{8}$$

From the Cauchy-Schwartz inequality,

$$\|\mathbf{R}\Delta\mathbf{h}_{k-1}^i\| \leq \|\mathbf{R}\|_F\|\Delta\mathbf{h}_{k-1}^i\|. \tag{9}$$

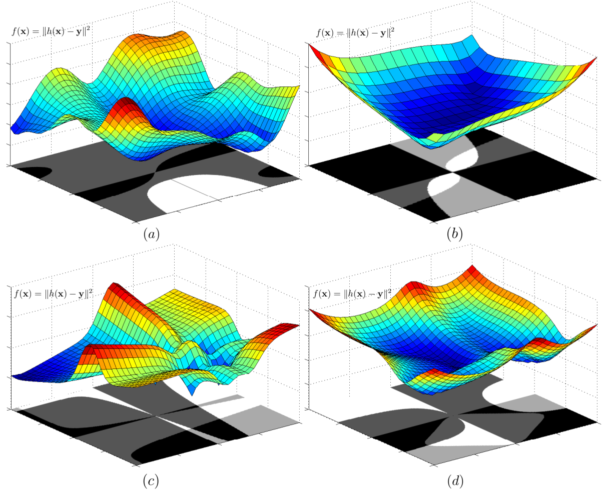


Figure 2. Illustration of DHD on four NLS functions where $\mathbf{h}(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}^2$. Different domains are colored in different grayscales.

Given the inequalities in Eqs. 8 and 9, the condition that makes Eq. 7 hold is,

$$\|\mathbf{R}\|_F = \sqrt{c}\|\mathbf{D}\|_F = \sqrt{c} \leq \frac{2}{K}\cos\theta^i. \tag{10}$$

Any $\mathbf{R} = c\mathbf{D}$ where $\sqrt{c} < \frac{2}{K}\min_i \cos\theta^i$ guarantees the inequality stated in Eq. 5. Therefore, there exists a generic DM for subset $S^{(0)}$. For other subsets in the partition a general choice of $\mathbf{D}$ has the entries

$$d_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ sign(\Delta x^i_{j,k-1}\Delta h^i_{j,k-1}) & \text{Otherwise.} \end{cases}$$

Following the same proof we can easily show DM exist for other subsets in the partition. □

Above, we proposed a simple partition strategy and proved the existence of DHD. In the next section, based on this strategy we derive a practical algorithm with applications to extrinsic camera calibration and facial feature tracking.

3. Application to Facial Feature Tracking

In this section we first review previous work on tracking profile-to-profile faces and the SDM's formulation on this problem. Next, we derive a simple strategy for finding DHD of the tracking objective function and following this strategy we extend SDM to track profile-to-profile faces.

3.1. Previous Work on Multi-view Face Tracking

Previous work on multi-view facial feature tracking can be grouped into two categories based on whether a 2D or 3D face model is used.

Let us first review some of the 2D model based approaches. The shape of a deformable object can be modeled by a probability density function. A multi-modal 2D face model can be represented either in a non-parametric way *e.g.*, kernel density estimation [38] or in a parametric way, *e.g.*, a mixture of Gaussians [8]. Therefore, there are two common strategies to extend traditional frontal face alignment methods to multi-view tracking. The first one is to build separate models according to the head pose. Some of the examples are multi-view Active Appearance Model (AAM) [10], view-based Active Wavelet Networks [19], and multi-view Direct Appearance Models [24]. The other is to use kernel methods. For example, Romdhani *et al.* [33] extended Active Shape Model [9] to track profile-to-profile faces. They used kernel PCA [36] to non-linearly model the shape variation across large pose changes. However, kernel-based density estimation is slow and its complexity increases with number of training samples. Another interesting work [47] treated the shape parameter and pose as hidden variables and framed the alignment problem into a Bayesian framework. However, the inference is intractable so the EM algorithm (local minima prone) is used to approximate the solution. Beyond the two common strategies, another way to address the multi-view problem would be online tracking. Ellis *et al.* [14] proposed an efficient online tracker using adaptive appearance models, and one could extend this approach to track faces and other nonrigid objects.

Next, let us take a look at 3D model based approaches. Matthews *et al.* [28] provided a detailed comparison between 2D and 3D face models in three different aspects, fitting speed, representational power, and construction. They concluded that 2D face model may be too "powerful" that can represent invalid faces. Xiao *et al.* [42] extended the AAM fitting algorithm to impose additional shape constraints introduced by a 3D model that are lacked in the 2D model. Baltrusaitis *et al.* [2] extended Constrained Local Models [11] for RGBD data streams and show better alignment performance than its original. However, the training data is difficult to collect. Gu and Kanade [17] formulated multi-view face alignment as a Bayesian inference problem with missing data, whose task is to solve 3D shape and 3D pose from the noisy and incomplete 2D shape observation. Recently, Cao *et al.* [5] extended an earlier 2D regression-based framework [6] with a 3D face model, but only near-frontal face results are shown in the experiments. Other interesting work [37, 40, 49] have been proposed for detecting facial landmarks in the profile-to-profile faces but they are not suitable for tracking applications. Note that most 3D based methods still rely on head pose to build separate models to address the multi-view problem.

Our work differs from existing approaches in several ways. First, our approach do not pre-build any shape or appearance model and we directly optimize over landmark coordinates. This has been shown to provide superior performance for facial feature tracking [43]. Second, our method provides a mathematically sound manner to partition the parameter space for facial feature tracking. Existing approaches typically find heuristic partition of the head pose angles. Finally, our method is general and can be applied to other problems, such as extrinsic camera calibration (see Section 4.2).

3.2. SDM's Formulation

Given an image $\mathbf{d} \in \mathbb{R}^{m\times 1}$ of m pixels, $\mathbf{d}(\mathbf{x}) \in \mathbb{R}^{p\times 1}$ indexes p landmarks in the image. $\mathbf{h}$ is a non-linear feature extraction function (*e.g.*, SIFT [27] or HoG [12]) and $\mathbf{h}(\mathbf{d}(\mathbf{x})) \in \mathbb{R}^{128p\times 1}$ in the case of extracting SIFT features. During training, we will assume that the correct p landmarks are known, and we will refer to them as $\mathbf{x}_*$. In this setting, SDM frames facial feature tracking as minimizing the following function over $\Delta\mathbf{x}$

$$f(\mathbf{x}_0 + \Delta\mathbf{x}) = \|\mathbf{h}(\mathbf{d}(\mathbf{x}_0 + \Delta\mathbf{x})) - \boldsymbol{\phi}_*\|_2^2, \tag{11}$$

where $\mathbf{x}_0$ is the initial configuration of the landmarks which corresponds to an average shape and $\boldsymbol{\phi}_* = \mathbf{h}(\mathbf{d}(\mathbf{x}_*))$ represents the SIFT values in the manually labeled landmarks. In the testing images, $\boldsymbol{\phi}_*$ is unknown. SDM modifies the objective to align with respect to the average template $\overline{\boldsymbol{\phi}}_*$ over all training images and the parameter update in Eq. 3 is modified accordingly,

$$\Delta\mathbf{x} = \mathbf{R}_k(\overline{\boldsymbol{\phi}}_* - \boldsymbol{\phi}_k). \tag{12}$$

SDM learns $\mathbf{R}_k$ by minimizing the loss between the true parameter update $\Delta\mathbf{x}_*^i = \mathbf{x}_*^i - \mathbf{x}_k^i$ and the expected one over all training samples

$$\sum_i \|\Delta\mathbf{x}_*^i - \mathbf{R}_k(\overline{\boldsymbol{\phi}}_* - \boldsymbol{\phi}_k^i)\|^2. \tag{13}$$

In training, SDM learns a sequence of DMs by iterating the above two steps, minimization of Eq. 13 and update 12 until convergence. In testing, those DMs are used for recursively updating shape parameters following Eq. 12.

3.3. Global SDM

SDM provides an efficient and accurate solution to track facial features in near-frontal faces, but it fails at tracking faces with large head rotations. When tracking profile-to-profile faces the shape parameter space is enlarged so it is unlikely to find a single valid DM (See section 2.1 and recall the two conditions for DM to exist). Theorem 1 shows that it is possible to partition the parameter space such that there exists a DM within each subset. Given a finite set of samples, finding the optimal DHD $S = \{S^t\}_1^T$ and its corresponding DMs $R = \{\mathbf{R}^t\}_1^T$ can be formulated as the following constrained optimization problem,

$$\min_{S,R} \sum_{t=1}^{T} \sum_{i\in S^t} \|\Delta\mathbf{x}_*^i - \mathbf{R}^t \Delta\boldsymbol{\phi}^{i,t}\|^2 \tag{14}$$

$$\text{s. t. } {\Delta\mathbf{x}_*^i}^\top \mathbf{R}^t \Delta\boldsymbol{\phi}^{i,t} > 0, \forall t, i \in S^t. \tag{15}$$

One can use a predefined T or choose the best T using a validation set. We denote $\overline{\boldsymbol{\phi}}_*^t - \boldsymbol{\phi}^i$ by $\Delta\boldsymbol{\phi}^{i,t}$, where $\overline{\boldsymbol{\phi}}_*^t$ is the template averaged over all image in the t^{th} subset. The constraints stated in 15 guarantee that $\mathbf{R}^t\mathbf{h}(\mathbf{x})$ is a monotone operator around $\mathbf{x}_*$, which is one condition ensuring that $\mathbf{R}^t$ is a generic DM within the t^{th} subset.

Minimizing 14 is NP-hard. We develop a deterministic approach to approximate the solution of 14. If $\mathbf{R}^t$ is a local minimizer, one necessary condition is that the partial derivative of 14 against $\mathbf{R}^t$ is zero yielding

$$\mathbf{R}^t = \Delta\mathbf{X}_*^t \Delta\Phi^{t\top} (\Delta\Phi^t \Delta\Phi^{t\top})^{-1}. \tag{16}$$

$\Delta\mathbf{X}_*^t$ and Φ^t are matrices whose columns are $\Delta\mathbf{x}_*^i$ and $\boldsymbol{\phi}^i$ from the t^{th} subset. Plugging Eq. 16 into the constraints in 15 yields,

$${\Delta\mathbf{x}_*^i}^\top \Delta\mathbf{X}_*^t \Delta\Phi^{t\top} (\Delta\Phi^t \Delta\Phi^{t\top})^{-1} \Delta\boldsymbol{\phi}^{i,t} > 0, \forall t, i \in S^t. \tag{17}$$

The sufficient conditions for 17 are

$${\Delta\mathbf{x}_*^i}^\top \Delta\mathbf{X}_*^t > \mathbf{0}, \forall t, i \in S^t \tag{18}$$

$$\Delta\Phi^{t\top} (\Delta\Phi^t \Delta\Phi^{t\top})^{-1} \Delta\boldsymbol{\phi}^{i,t} > \mathbf{0}, \forall t, i \in S^t \tag{19}$$

From the fact that any two vectors within the same hyperoctant (the generalization of quadrant) have a positive dot product, we design a partition such that each subset occupies a hyperoctant in the parameter space. This partition satisfies the inequalities in 18. We can apply the same strategy to further partition each subset according to the hyperoctants in feature space, which yields the following inequalities

$$\Delta\Phi^{t\top} \Delta\boldsymbol{\phi}^{i,t} > \mathbf{0}, \forall t, i \in S^t \tag{20}$$

The covariance matrix $\Phi\Phi^\top$ is positive-definite (if not, a diagonal matrix can be added). The inverse of a positive definite matrix is also positive definite. This fact along with 20 suffice to show the inequalities in 19. However, this partition is impractical leading to exponential number of DMs so we propose the following approximation.

In the case of human faces, $\Delta\mathbf{x}$ and $\Delta\boldsymbol{\phi}$ are embedded in a lower dimensional manifold. We perform dimension reduction (PCA) on the whole training set $\Delta\mathbf{X}$ and project the data onto the subspace expanded by the first two most

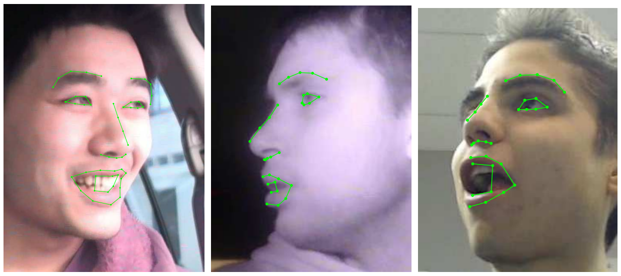


Figure 3. Three sample images from the Driver-face dataset. a) A near-frontal face labeled with 49 points. The subject is recorded inside of a car during daytime. b) A profile face labeled with 31 points. The subject is recorded during nighttime under IR light. c) The subject is recorded indoors.

dominant directions. This gives us a partition in $\mathbb{R}^2$ where each subset occupies a quadrant. Each subset inside this partition is further partitioned into two halves based on the first principle component learned from $\Delta\Phi$. This partition strategy gives us eight subsets so eight DMs are learned in each iteration of the algorithm. The PCA bases are saved and used to determine which DM to use in testing time. The training of GSDM converges in four iterations. In testing $\mathbf{x}_*$ is unknown and assuming that the movement between two consecutive frames is small the prediction of the previous frame is used to approximate $\Delta\mathbf{x}_*$. We only used two PCA bases, although one can increase the number of bases to create more subsets in the partition. The approximation would be more accurate at the same time more training data will be needed to learn a reliable DM. One can also use nonlinear dimension reduction techniques [32]. This simple partition strategy has been validated in our experiments and yields promising results.

4. Experiments

This section illustrates the effectiveness of GSDM on two computer vision problems. First, we illustrate how GSDM is able to track the face from profile to profile. Second, we show how GSDM can be applied to solve the extrinsic camera calibration problem.

4.1. Facial Feature Tracking from Profile to Profile

Over the past few years, researchers in the face alignment field have made rapid progress on improving the landmark accuracy and speed of the algorithms. Such progress is made possible by the availability of larger and more challenging datasets *e.g*., LFPW [3], Helen [23], AFLW [21], AFW [35], IBUG [34]. However, there is a lack of datasets for evaluation of face tracking from profile to profile as well as a standard protocol for evaluating tracking performance. To fill the void, we build two challenging datasets, Distracted Driver Face(DDF) and Naturalistic Driving Study(NDS), and propose a standard evaluation protocol for facial feature tracking. Both the evaluation protocol code (including the labels) are made available for the research community[2].

The **DDF dataset** contains 15 video sequences, a total of 10,882 frames. Each sequence captures a single subject performing distracted driving in a stationary vehicle or an indoor environment. 12 out of 15 videos are recorded with subjects sitting inside of a vehicle. Five of them are recorded in the night under infrared (IR) light and the others are recorded during the daytime under natural lighting. The remaining three are recorded indoors. Fig. 3 shows one example of each category and its corresponding labels.

The **NDS dataset** [41] contains 20 subsequences of driver faces recorded during a drive conducted between the Blacksburg, VA and Washington, DC areas. Each sequence consists of a one-minute video recorded at 15 fps with a resolution of 360 × 240. For both datasets, we labeled one in every ten frames and each labeled frame consists of either 49 landmarks (near-frontal faces) or 31 landmarks (profile faces). Both datasets consist of many faces with extreme pose (±90° yaw, ±50° pitch) and many under extreme lighting condition (*e.g*., IR). NDS is the more challenging one due to the low spatial and temporal resolution.

Evaluation protocol: A popular evaluation metric for facial feature detection is the cumulative error curve. However, this curve cannot take into account the frames that are lost during tracking. We propose the Cumulative Error Histogram (CEH) as the evaluation metric. The idea of CEH is to quantize the tracking error at different scales. The histogram will have k bins, where the i^{th} bin counts the fraction of frames (number of frames over total number of frames) with errors less than the i^{th} error scale. For the frames where the tracker is lost or the landmark error is larger than the last error scale, we add them to the last bin. For the successfully tracked frames, the error is measured using the normalized root mean square (RMS) metric. In previous work, normalization is often done by using the inter-ocular distance. However, for a profile face such distance tends to go to zero so we use the face length as a reference approximated by the distance between the lower lip point and the inner eyebrow point. The mean of all bins in a CEH can be used as single-value score to compare among different tracking methods. The CEH score has the value between 1 and $\frac{1}{k}$ with higher value indicating better performance. In the worst case, *e.g*., no face is tracked in a sequence, all bins except the last one equal to zero yielding a score of $\frac{1}{k}$. On the other hand, the score equals one if all frames fall in the first bin.

In the experiments, both the SDM and GSDM algorithms are trained on MPIE [16] and a subset of LFW [20]. We used CEH to measure the performance of each tracker, and

[2] http://humansensing.cs.cmu.edu/xxiong

$k = 10$ and the max error is set to be 0.06. A face detector (OpenCV [4] in our case) is called once the tracker is lost and the tracker is not re-initialized until a valid face is detected. No manual effort is involved to re-initialize both trackers. Fig. 4 shows CEHs between SDM and GSDM in both datasets. GSDM is able to track more frames and provides more accurate landmark prediction than SDM. Both algorithms have significant performance drop-off in NDS dataset because of the noisy, low resolution images and heavy occlusion introduced by the sunglasses. Additionally, images in NDS dataset are significantly different than the ones in our training set. Example results can be found in Fig. 6 or from the link below[3]. Our C++ implementation averages around 8ms per frame, tested with an Intel i7 3752M processor.

4.2. Extrinsic Camera Calibration

This section reports the experimental results on extrinsic camera calibration using GSDM and a comparison with SDM and the widely popular POSIT method [13]. For both SDM and GSDM, extrinsic camera calibration is formulated as minimizing the following NLS function,

$$\min_{\mathbf{x}} \|\mathbf{h}(\mathbf{x}, \mathbf{M}) - \mathbf{U}\|_F,$$

where $\mathbf{h}$ is the projection function and $\mathbf{x} = [\boldsymbol{\theta}; \mathbf{t}]$, $\boldsymbol{\theta}, \mathbf{t}$ are vectors of three rotation angles and translations, respectively. $\mathbf{M} \in \mathbb{R}^{3\times n}$ is the 3D object in consideration, and $\mathbf{U} \in \mathbb{R}^{2\times n}$ is the image projection under the pose parameter $\mathbf{x}$. In the training of GSDM, we follow a similar partition strategy introduced in section 3.3. Each dimension in the parameter space is independent of each other so no dimension reduction is needed. DHD are found by splitting the parameter space according to three rotation angles. Each domain within DHD occupies an octant in $\mathbb{R}^3$. It gives us eight DMs to learn in every iteration and training iteration is set to be 10. In testing, unlike in the tracking application where we can use the previous frame information as an approximation of $\mathbf{x}_*$, we iterate through all DMs and uses the one that returns the minimum reprojection error.

The experiment is set up as follows. We select three different 3D objects: a cube, a face, and a human body[4] (see Fig. 5a). We place a virtual camera at the origin of the world coordinates. In this experiment, we set the focal length (in terms of pixels) to be $f_x = f_y = 1000$ and principle point to be $[u_0, v_0] = [500, 500]$. The skew coefficient is set to be zero. The training and testing data are generated by placing a 3D object at $[0, 0, 2000]$, perturbed with different 3D translations and rotations. The POSIT algorithm does not require labeled data. Three rotation angles are uniformly sampled from $-60°$ to $60°$ with increments of $10°$ in training and $7°$ in testing. Three translation values are uniformly sampled from -400mm to 400mm with increments of 200mm in training and 170mm in testing. Then, for each combination of the six values, we compute the object's image projection using the above virtual camera and use it as the input for both algorithms. White noise ($\sigma^2 = 4$) is added to the projected points. In our implementation of both SDM and GSDM, to ensure numerical stability, the image coordinates $[u, v]$ of the projection are normalized as follows: $\begin{bmatrix}\hat{u}\\ \hat{v}\end{bmatrix} = \begin{bmatrix}(u - u_0)/f_x\\ (v - v_0)/f_y\end{bmatrix}$.

Fig. 5b shows the mean errors and standard deviations of the estimated rotations (in degrees) and translations (in mm) for three algorithms. SDM performs the worst among the three because the parameter space is so large that there not exists a single DM. GSDM overcomes this problem by partitioning the large space into eight subsets and learning eight DMs. Both GSDM and POSIT achieve around $1°$ accuracy for rotation estimation, but GSDM is much more accurate for translation. This is because POSIT assumes a scaled orthographic projection, while the true image points are generated by a perspective projection.

5. Conclusions

SDM provides an elegant and efficient method to solve local optimization problems in NLS functions. However, SDM is a local algorithm and it is likely to average conflicting gradient directions. This paper proposed GSDM, an extension of SDM that divides the search space into domains of similar gradient directions. We illustrated its effectiveness in two applications, facial feature tracking and extrinsic camera calibration. In both applications, we demonstrated GSDM's superior performance to previous methods. However, our partition strategy is a more natural fit for tracking applications since an approximate $\mathbf{x}_*$ is needed. In the case of extrinsic camera calibration, no previous frame information is given so we have to iterate through all DMs. In the experiment of camera calibration, we made one implicit assumption that during optimization the updating parameter never goes out of the domain initially selected. In other applications this assumption may not hold, so in the worst case we may need to iterate through an exponential number of combinations of DMs before finding the optimal solution. It is possible that better partition strategies exist within the GSDM framework, and we will explore those in the future work. Besides GSDM, we established the connection between SDM and Imitation Learning. In addition, we built a public dataset and proposed an evaluation protocol for benchmarking facial feature tracking methods.

[3] http://goo.gl/EGiUFV

[4] www.robots.ox.ac.uk/~wmayol/3D/nancy_matlab.html

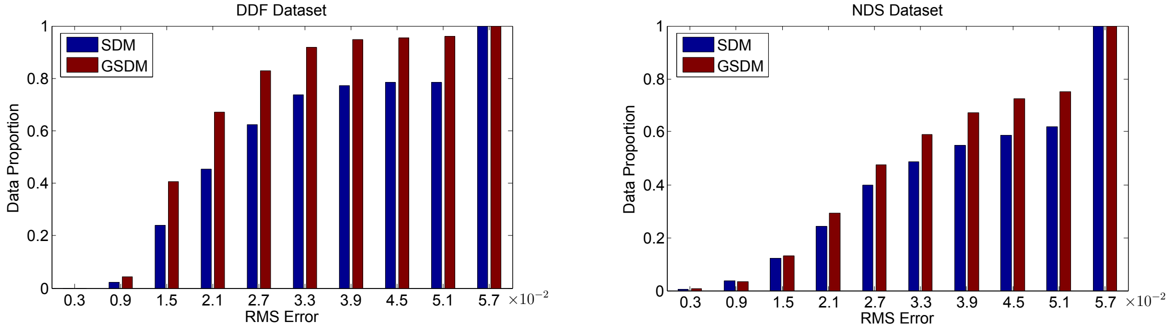
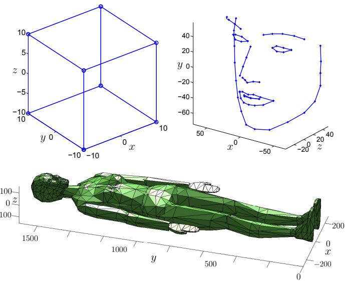


Figure 4. Performance comparison between SDM and GSDM in terms of CEH on DDF dataset (left) and NDS dataset (right).



(a)

		θ_x	θ_y	θ_z	t_x	t_y	t_z
Cube	SDM	5.2 ± 5.2	8.9 ± 8.4	10.5 ± 8.9	18.0 ± 14.4	18.4 ± 14.2	129.9 ± 120.9
	GSDM	$\mathbf{0.7 \pm 0.7}$	0.9 ± 0.9	0.7 ± 0.7	$\mathbf{2.4 \pm 3.1}$	$\mathbf{2.4 \pm 3.1}$	$\mathbf{17.1 \pm 17.3}$
	POSIT	0.8 ± 0.7	$\mathbf{0.9 \pm 0.8}$	$\mathbf{0.7 \pm 0.6}$	66.5 ± 48.1	66.3 ± 51.3	69.4 ± 50.3
Face	SDM	5.8 ± 6.2	10.3 ± 10.4	10.9 ± 12.4	14.6 ± 18.1	14.3 ± 18.7	123.4 ± 132.9
	GSDM	$\mathbf{0.8 \pm 1.0}$	$\mathbf{1.1 \pm 1.2}$	$\mathbf{0.9 \pm 1.0}$	$\mathbf{2.5 \pm 8.3}$	$\mathbf{2.4 \pm 7.8}$	$\mathbf{18.9 \pm 19.8}$
	POSIT	1.5 ± 1.3	1.9 ± 1.7	1.5 ± 1.3	28.6 ± 24.4	32.5 ± 22.8	47.3 ± 36.8
Body	SDM	3.0 ± 4.4	4.4 ± 6.1	4.7 ± 6.9	12.1 ± 20.5	12.3 ± 20.7	101.3 ± 134.2
	GSDM	$\mathbf{0.6 \pm 0.1}$	$\mathbf{0.7 \pm 0.1}$	$\mathbf{0.8 \pm 0.1}$	$\mathbf{0.3 \pm 0.8}$	$\mathbf{0.3 \pm 0.9}$	$\mathbf{1.5 \pm 3.9}$
	POSIT	0.6 ± 0.6	2.5 ± 2.6	1.1 ± 0.9	38.0 ± 22.1	28.5 ± 27.2	37.8 ± 30.9

(b)

Figure 5. a) 3D objects used in the experiments of extrinsic camera calibration. Units are in millimeters (mm). b) Experimental results on extrinsic camera calibration in terms of mean errors and their standard deviations from three algorithms. Rotation errors are measured in degrees and translation errors are measured in mm.

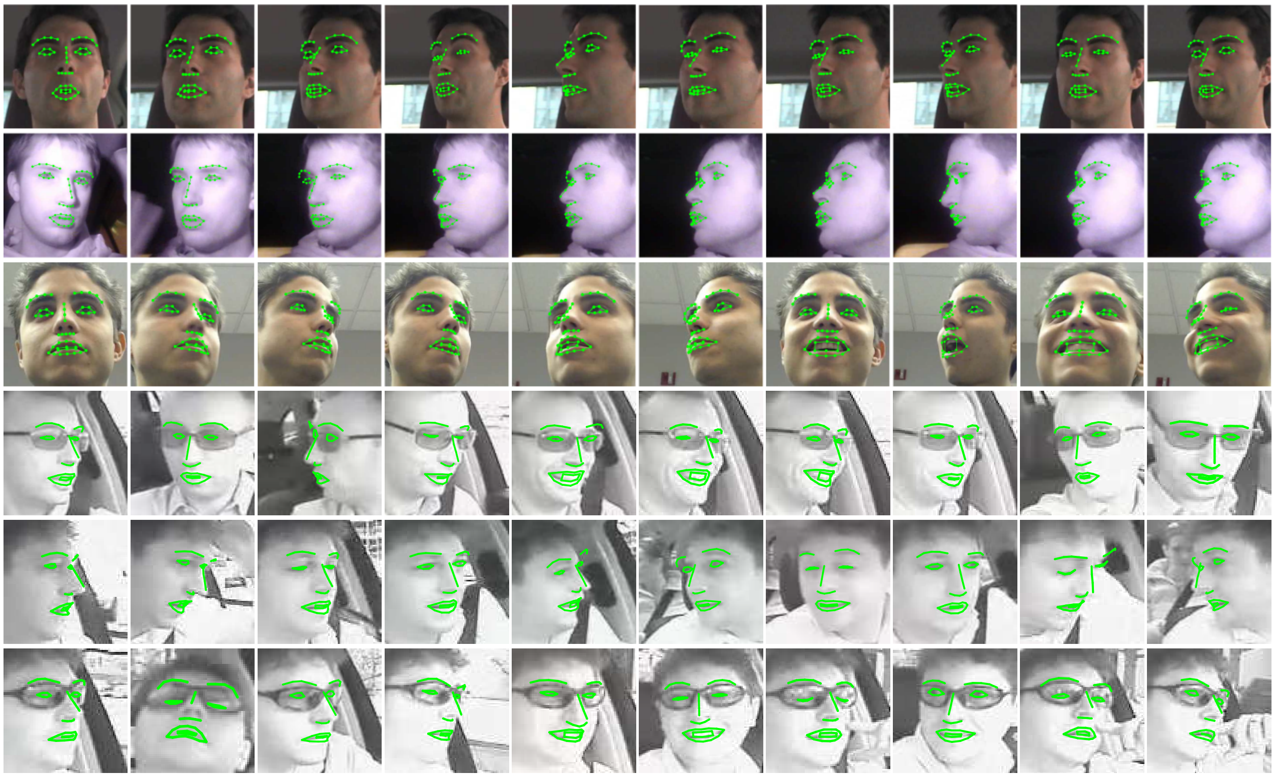


Figure 6. Tracking results from GSDM on the DDF dataset (top three rows) and NDS dataset (bottom three rows).

Acknowledgements

This work was partially supported by the Federal High Way Administration (FHWA) award number , DTFH61-14-C-00001. The findings and conclusions of this paper are those of the author(s) and do not necessarily represent the views of the VTTI, SHRP 2, the Transportation Research Board, FHWA, or the National Academies.

References

[1] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1859–1866, 2014. 2

[2] T. Baltrusaitis, P. Robinson, and L. Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2610–2617, 2012. 4

[3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011. 6

[4] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 7

[5] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (TOG)*, 33(4):43, 2014. 4

[6] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012. 4

[7] W.-S. Chu, F. Zhou, and F. De la Torre. Unsupervised temporal commonality discovery. In *ECCV*, 2012. 1

[8] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–573, 1999. 4

[9] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995. 4

[10] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. View-based active appearance models. *Image and vision computing*, 20(9):657–664, 2002. 4

[11] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Journal of Pattern Recognition*, 41(10):3054–3067, 2008. 4

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005. 5

[13] D. F. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. *IJCV*, 15:123–141, 1995. 7

[14] L. Ellis, N. Dowson, J. Matas, and R. Bowden. Linear regression and adaptive appearance models for fast simultaneous modelling and tracking. *International journal of computer vision*, 95(2):154–179, 2011. 4

[15] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. AMS, Providence, RI, 1986. 1

[16] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. In *AFGR*, 2007. 6

[17] L. Gu and T. Kanade. 3d alignment of face in a single image. In *Computer Vision and Pattern Recognition*, volume 1, pages 1305–1312, 2006. 4

[18] N. Hansen, S. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation*, 11(1):1–18, 2003. 1

[19] C. Hu, R. Feris, and M. Turk. Real-time view-based face alignment using active wavelet networks. In *Analysis and Modeling of Faces and Gestures*, pages 215–221, 2003. 4

[20] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 6

[21] M. Kostinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Computer Vision Workshops (ICCV Workshops)*, pages 2144–2151, 2011. 6

[22] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *Pattern Analysis and Machine Intelligence*, 31(12):2129–2142, 2009. 1

[23] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, pages 679–692, 2012. 6

[24] S. Z. Li, H. Zhang, Q. Cheng, et al. Multi-view face alignment using direct appearance models. In *Automatic Face and Gesture Recognition*, pages 324–329, 2002. 4

[25] L. Liu, J. Hu, S. Zhang, and W. Deng. Extended supervised descent method for robust face alignment. In *ACCV*, 2014. 2

[26] C. Long, X. Wang, G. Hua, M. Yang, and Y. Lin. Accurate object detection with location relaxation and regionlets relocalization. In *ACCV*, 2014. 2

[27] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 5

[28] I. Matthews, J. Xiao, and S. Baker. 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. *International journal of computer vision*, 75(1):93–113, 2007. 4

[29] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110, 2003. 1

[30] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004. 1

[31] J. Pascual Starink and E. Backer. Finding point correspondences using simulated annealing. *Pattern Recognition*, 28(2):231–240, 1995. 1

[32] R. Pless and R. Souvenir. A survey of manifold learning for images. *IPSJ Transactions on Computer Vision and Applications*, 1:83–94, 2009. 6

[33] S. Romdhani, S. Gong, A. Psarrou, et al. A multi-view nonlinear active shape model using kernel pca. In *BMVC*, volume 10, pages 483–492, 1999. 4

[34] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Computer Vision Workshops (ICCVW)*, pages 397–403, 2013. 6

[35] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 896–903, 2013. 6

[36] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *Artificial Neural Networks-CANN'97*, pages 583–588. Springer, 1997. 4

[37] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3460–3467. IEEE, 2013. 4

[38] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986. 4

[39] C. Sminchisescu and B. Triggs. Building roadmaps of local minima of visual models. In *ECCV*, pages 566–582, 2002. 1

[40] B. M. Smith, J. Brandt, Z. Lin, and L. Zhang. Nonparametric context modeling of local appearance for pose-and expression-robust facial landmark localization. In *CVPR*, 2014. 4

[41] Transportation Research Board of the National Academies of Science. The 2nd strategic highway research program naturalistic driving study dataset. Available from the SHRP 2 NDS InSight Data Dissemination web site: https://insight.shrp2nds.us/, 2013. 6

[42] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+ 3d active appearance models. In *CVPR*, pages 535–542, 2004. 4

[43] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR)*, pages 532–539, 2013. 2, 5

[44] X. Xiong and F. De la Torre. Supervised descent method for solving nonlinear least squares problems in computer vision. *arXiv preprint arXiv:1405.0601*, 2014. 2

[45] J. Yan, Z. Lei, Y. Yang, and S. Z. Li. Stacked deformable part model with shape regression for object part localization. In *ECCV*, pages 568–583, 2014. 2

[46] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *ECCV*, pages 1–16, 2014. 2

[47] Y. Zhou, W. Zhang, X. Tang, and H. Shum. A bayesian mixture model for multi-view face alignment. In *CVPR*, 2005. 4

[48] S. Zhu, C. Li, C. C. Loy, and X. Tang. Transferring landmark annotations for cross-dataset face alignment. *arXiv preprint arXiv:1409.0602*, 2014. 2

[49] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 4