Technical Report
TTIC-TR-2011-2

April 2011

Pose Estimation Under Anisotropic Scaling with Automatic Initialization

Payman Yadollahpour
TTI-Chicago
pyadolla@ttic.edu

Gregory Shakhnarovich
TTI-Chicago
greg@ttic.edu

ABSTRACT

In this paper we investigate solutions to the problem of pose estimation for a class of rigid objects parameterized by anisotropic scaling (variable aspect ratios). We describe extensions of two known pose estimation algorithms to this scenario, that allow them to estimate aspect, in addition to rotation and translation. Our first contribution is a fully automatic example-based initialization scheme. We show that this scheme benefits iterative pose estimation methods across the board, dramatically improving convergence to accurate results. Our second contribution is a pose estimation algorithm called Scaled Exterior Orientation (SEO) that handles anisotropic scaling. In experiments on two object classes we show that combined with the proposed automatic initialization, SEO achieves best results in low noise conditions, and is on par with other algorithms under severe noise.

1 Introduction

There are many reasons why the problem of pose estimation is important. The task of estimating as precisely as possible the orientation of 3D objects is an important component of planning in robotics - whether by an autonomous navigation when one needs to avoid obstacles, or in a grasping scenario. There has also been increasing interest in integrating pose estimation into object detection systems [13, 12, 1]. Our interest in the problem was in fact motivated by the latter application. Many interesting semantic categories correspond to families of 3D objects that, at least approximately, have the same basic shape, but with varying aspect ratios. For instance, cars can be very long (limousines) or short (a Smartcar), tall (Jeep) or low (a convertible) relative to their length.

This paper is about the problem of pose estimation for such families of objects. We represent the changing aspect by anisotropic scaling applied to a fixed-aspect canonical model. Two example families we consider here are box-shaped objects and primitive car models. We assume that the system is given a set of points in a single image, corresponding to a sufficient subset of keypoints on the object. We also assume that it is known which pairs of the input points form line segments corresponding to projections of edges. Our goal is to recover, from these image features, full information about the objects pose (rotation and translation) and shape (here just the aspect ratios). The absolute scale, of course, is ambiguous when only one image is presented, and so we need to estimate the total of eight DOF: three for rotation, three for translation, assuming fixed scale, and two for anisotropic scaling (aspect).

Problem setup We represent the object class of interest by a canonical model - a class prototype. The model consists of a set of V 3D vertices, and E edges. Each edge is defined by two vertices it connects. Each edge is also associated with two faces on which it is incident, and thus with two surface normals pointing from the object out. This information is necessary when we need to reason about visibility of points on the object. Note that we do not assume convexity of the object, and indeed the car class used in our experiments is non-convex. The vertices and normals of the canonical model are represented in object-centered coordinate system. The origin is in the object centroid, and the canonical rotation aligns the object in a certain way with the axes of 3D space.

Roadmap In the next section we describe prior work on pose estimation of rigid bodies, focusing on the methods considered state of the art. We then describe in Section 3 extensions of these methods to anisotropic scaling scenario. Section 4 is devoted to the novel example-based initialization method, which proves to improve accuracy of estimation methods across the board in experiments, reported in Section 5.

2 Background

The problem of pose estimation is known as *exterior orientation* (EO) problem: finding the 3D transformation that best aligns a set of 3D points to a set of 2D image points under perspective projection, given point correspondences.

Given a set of 3D points $\mathbf{p}_i = [x_i, y_i, z_i]^T$, $i = 1, \ldots, n$ represented in the object coordinate system, the corresponding camera coordinate representation is given as,

$$\mathbf{q}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t} = [x'_i, y'_i, z'_i]^T. \tag{1}$$

In the camera reference frame the center of projection of the camera is at the origin and the camera is pointed in the positive z-axis direction, with the image plane at $z' = 1$. A 3D point $\mathbf{p}_i$ is projected onto a normalized image point $\mathbf{v}_i = [u_i, v_i, 1]^T$ in the image plane.

What we are given are observed values $\hat{\mathbf{v}}_i = [\hat{u}_i, \hat{v}_i, 1]^T$. We will assume for now that the object-to-image point correspondences, $\mathbf{p}_i \leftrightarrow \hat{\mathbf{v}}_i$, are known. The standard method for solving the EO problem is to minimize the squared

error between the projected object points and the observed image points [8],

$$(\mathbf{R}^*, \mathbf{t}^*) = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \quad \sum_{i=1}^{n} \left[(\hat{u}_i - u_i)^2 + (\hat{v}_i - v_i)^2 \right] \tag{2}$$
$$\text{s.t.} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}_3, \quad \det(\mathbf{R}) = 1$$

The objective (2) is usually minimized using Gauss-Newton or Levenberg-Marquardt methods. The latter behaves better at avoiding local minima, yet global convergence is not guaranteed and a good initial guess is usually required to converge to the correct solution.

Another common set of solutions rely on removing the orthogonality constraints on $\mathbf{R}$ in (2), then finding the solution that minimizes the objective, followed by orthogonalization of the solution. One such method is the Direct Linear Transform (DLT).

2.1 Estimation via DLT

The DLT algorithm [5] is applied when, in addition to correspondences $\mathbf{p}_i \leftrightarrow \hat{\mathbf{v}}_i$, the intrinsic camera parameter matrix $\mathbf{K}$ is known. The objective of DLT is to estimate the camera matrix, $\mathbf{H} \in \mathbb{R}^{3\times4}$, such that,

$$\mathbf{v}_i \simeq \mathbf{H}\mathbf{p}_i, \quad \text{for } i = 1, \ldots, n. \tag{3}$$

Expressing the constraints in (3) in equivalent cross product form, denoting the rows of $\mathbf{H}$ by $\mathbf{h}_1, \ldots, \mathbf{h}_3$, and writing the homogenous image points as $\mathbf{v}_i = (u_i, v_i, w_i)^T$, we get a linear system. The least squares solution of this system provides an estimate of $\mathbf{H}$, and can be obtained with at least six correspondences.

The DLT algorithm is not invariant to similarity transformation, and so it is essential to apply a normalizing transformations ($\mathbf{U}$ and $\mathbf{T}$), or pre-conditioning, to both $\{\mathbf{p}\}$ and $\{\mathbf{v}\}$. Furthermore, in practice the DLT solution can be further refined by using it as a starting point, and minimizing the geometric error (reprojection error in the image). This optimization can be performed using an iterative algorithm such as Levenberg-Marquardt. Finally, the camera matrix in the original coordinate system can thus be obtained as, $\mathbf{H}^* = \mathbf{T}^{-1}\tilde{\mathbf{H}}^*\mathbf{U}$. We can thus decompose $\mathbf{H}^* = \left[\widehat{\mathbf{R}}|\widehat{\mathbf{t}}\right]$. Note that DLT does not enforce orthogonality constraints on $\mathbf{R}$, and the estimated $\widehat{\mathbf{R}}$ must be orthonormalized to obtain a valid rotation matrix.

2.2 Orthogonal Iteration Algorithm

Yet another way to approach the EO problem is to minimize error in the object space instead of in the image plane [9]. We describe it here in some detail, since our work extends it to the anisotropic scaling setup. The goal is to minimize the following error vector,

$$\mathbf{e}_i = (\mathbf{I}_3 - \hat{\mathbf{V}}_i)(\mathbf{R}\mathbf{p}_i + \mathbf{t}), \tag{4}$$

over all observed points $i = 1, \ldots, n$, where $\hat{\mathbf{V}}_i = \frac{\hat{\mathbf{v}}_i\hat{\mathbf{v}}_i^T}{\hat{\mathbf{v}}_i^T\hat{\mathbf{v}}_i}$ is the operator that projects a 3D point in camera coordinates onto the ray from camera center through $\hat{\mathbf{v}}_i$.

This is equivalent to minimizing sum of squared error over all data points,

$$\min_{\mathbf{R},\mathbf{t}} \quad \sum_{i=1}^{n} ||(\mathbf{I}_3 - \hat{\mathbf{V}}_i)(\mathbf{R}\mathbf{p}_i + \mathbf{t})||_2^2 \tag{5}$$
$$\text{s.t.} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}_3, \quad \det(\mathbf{R}) = 1 \tag{6}$$

where $\mathbf{I}_3$ is the 3×3 identity. Since (5) is quadratic in $\mathbf{t}$, the optimal translation in terms of $\mathbf{R}$ can be written [9] in the following way:

$$\mathbf{t}^*(\mathbf{R}) = \frac{1}{n}\left(\mathbf{I}_3 - \frac{1}{n}\sum_{j=1}^{n}\hat{\mathbf{V}}_j\right)^{-1}\sum_{j=1}^{n}(\hat{\mathbf{V}}_j - \mathbf{I}_3)\mathbf{R}\mathbf{p}_j. \tag{7}$$

We can now define $\mathbf{q}_i(\mathbf{R}) \triangleq \hat{\mathbf{V}}_i(\mathbf{R}\mathbf{p}_i + \mathbf{t}(\mathbf{R}))$, giving an equivalent form for (5), under the same constraints (6):

$$\min_{\mathbf{R}} \quad \sum_{i=1}^{n} ||\mathbf{R}\mathbf{p}_i + \mathbf{t}(\mathbf{R}) - \mathbf{q}_i(\mathbf{R})||_2^2 \tag{8}$$

This is very similar to the definition of *absolute orientation* (AO) problem, which is about finding 3D transformation that aligns two sets of 3D points. Intuitively, we seek rotation of the canonical model that matches the rays passing through observed image points.

There exists a closed form solution to the AO problem [6, 7]. However, as shown in [9], it can not be applied to (8) since the $\mathbf{q}$s are dependent on $\mathbf{R}$. The problem can instead be solved iteratively as follows. Given the soluion $\mathbf{R}^{(k)}$ in iteration k, we can compute $\mathbf{t}^{(k)} = \mathbf{t}(\mathbf{R}^{(k)})$, and $\mathbf{q}_i^{(k)} = \mathbf{R}^{(k)}\mathbf{p}_i + \mathbf{t}^{(k)}$. We can then estimate $\mathbf{R}^{(k+1)}$ by solving,

$$\begin{aligned}\mathbf{R}^{(k+1)} &= \underset{\mathbf{R}:\,\mathbf{R}^T\mathbf{R}=\mathbf{I},\det(\mathbf{R})=1}{\operatorname{argmin}} \sum_{i=1}^{n} ||\mathbf{R}\mathbf{p}_i + \mathbf{t} - \hat{\mathbf{V}}_i\mathbf{q}_i^{(k)}||_2^2 \\ &= \underset{\mathbf{R}:\,\mathbf{R}^T\mathbf{R}=\mathbf{I},\det(\mathbf{R})=1}{\operatorname{argmax}} \operatorname{Tr}\left(\mathbf{R}^T\mathbf{M}\mathbf{R}^{(k)}\right),\end{aligned} \tag{9}$$

where $\mathbf{M}$ is the cross-covariance matrix of $\{\mathbf{q}\}$ and $\{\mathbf{p}\}$. The solution to (9) can be obtained via SVD: if $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, we have $\mathbf{R}^{(k+1)} = \mathbf{V}\mathbf{U}^T$. This process is repeated until a stationary point is reached.

2.3 SOFTPosit

In the SOFTPosit system [3] an EM algorithm, which iterates between updating the pose estimate and solving for optimal correspondences under the current pose. The algorithm can be guided by an initial setting of $\mathbf{S}$ and $\mathbf{t}$, and proceeds by establishing correspondences using weak perspective, refined by estimating new $\mathbf{S}$ and $\mathbf{t}$.

3 Extensions to anisotropic scaling

The survey of existing methods in the preceding section identified three promising approaches described in the literature for estimating pose of fully rigid objects: DLT, Orthogonal Iteration Algorithm, and SOFTPosit. Here we describe our extension of the former two methods to handle anisotropic scaling. We could not find a straightforward way to modify SOFTPosit; however, it turns out that our automatic initialization method described in Section 4 allows applying SOFTPosit in its original form even when input is affected by anisotropic scaling.

Mathematically, (1) is replaced with

$$\mathbf{q}_i = \mathbf{R}\mathbf{S}\mathbf{p}_i + \mathbf{t} = [x_i', y_i', z_i']^T, \tag{10}$$

where the diagonal matrix $\mathbf{S}$ describes the aspect ratios achieved by anisotropically scaling the canonical model. We will assume, without loss of generality, that the largest element of $\mathbf{S}$ is precisely 1.

3.1 The DLT-Fact algorithm

The only change to the algorithm in Section 2.1 is in the decomposition of $\mathbf{H}^*$. Assuming anisotropic scaling is equivalent to assuming that $\mathbf{H}^* = [\mathbf{RS}|\mathbf{t}]$, where $\mathbf{S} = \text{diag}(s_1, s_2, s_3) \in \mathbb{R}^{3\times3}$. Let $\mathbf{U} = \mathbf{RS} \in \mathbb{R}^{3\times3}$. We can factorize $\mathbf{U} = \mathbf{R}'\mathbf{S}'$ using QR-decomposition [10], resulting in orthogonal $\mathbf{R}'$ and upper triangular $\mathbf{S}'$. Under noisy observations, deviation of $\mathbf{S}'$ from diagonal could be significant. We therefore compute $\mathbf{S}'' = \text{diag}(\text{diag}(\mathbf{S}'))$ (removing off-diagonal entries), and re-estimate rotation $\mathbf{R}'' = \mathbf{U}\left(\mathbf{S}''\right)^{-1}$. Finally, we orthogonalize it, and obtain estimates $\widehat{\mathbf{R}}_0 = \text{orth}\left(\mathbf{R}''\right)$, $\widehat{\mathbf{S}}_0 = \mathbf{S}''$. Following [5], these estimates are refined using nonlinear least squares (Levenberg-Marquardt), yielding $\widehat{\mathbf{R}}$ and $\widehat{\mathbf{S}}$.

3.2 Scaled Exterior Orientation (SEO)

We now describe the modified orthogonal iteration algorithm from Section 2.2, extended to handle anistropic scaling. We call it the Scaled Exterior Orientation algorithm (SEO). In our experiments reported in 5 we found this algorithm to be on par with or superior to others, in particular in low noise conditions.

Our objective (5) of minimizing error in 3D becomes

$$\begin{aligned} \min_{\mathbf{S},\mathbf{R},\mathbf{t}} \quad & \sum_{i=1}^{n} \left|\left| \left(\mathbf{I}_3 - \hat{\mathbf{V}}_i\right)\left(\mathbf{RSp}_i + \mathbf{t}\right) \right|\right|_2^2 \\ \text{s.t.} \quad & \mathbf{R}^T\mathbf{R} = \mathbf{I}_3, \quad \det(\mathbf{R}) = 1 \\ & \mathbf{S} = \text{diag}(s_1, \ldots, s_m), \quad s_j \in (0, 1] \end{aligned} \tag{11}$$

The derivation of the optimal translation follows closely the derivation of (7) in [9]; we omit the details here, and only state the result:

$$\mathbf{t}^*(\mathbf{R}, \mathbf{S}) = \frac{1}{n}\left(\mathbf{I}_3 - \frac{1}{n}\sum_{j=1}^{n}\hat{\mathbf{V}}_j\right)^{-1}\sum_{j=1}^{n}\left(\hat{\mathbf{V}}_j - \mathbf{I}_3\right)\mathbf{RSp}_j. \tag{12}$$

Plugging (12) into (11), we get the folowing function to be minimized w.r.t. $\mathbf{R}, \mathbf{S}$:

$$F(\mathbf{S}, \mathbf{R}) = \sum_{i=1}^{n}\left|\left| \left(\mathbf{RSp}_i + \mathbf{t}^*(\mathbf{R}, \mathbf{S}) - \mathbf{q}_i(\mathbf{R}, \mathbf{S})\right)\right|\right|_2^2, \tag{13}$$

where $\mathbf{q}_i(\mathbf{R}, \mathbf{S}) = \hat{\mathbf{V}}_i\big(\mathbf{RSp}_i + \mathbf{t}\left(\mathbf{R}, \mathbf{S}\right)\big)$. Let $\mathbf{p}'_i$ and $\mathbf{q}'_i$ be the values after we shift both $\mathbf{p}$s and $\mathbf{q}$s to be centered at origin. We then get

$$F(\mathbf{R}, \mathbf{S}) = \sum_{i=1}^{n} ||\mathbf{RSp}'_i - \mathbf{q}'_i||_2^2. \tag{14}$$

This is an instance of the AO problem, but under unknown anisotropic scaling. To our knowledge no closed form solution to this problem exists, but an algorithm called Scaled Iterative Closest Point (SICP) [4] solves this iteratively. SICP, which is a generalization of the well known (unscaled) ICP [2], in fact solves a more complex problem of finding the transformation under unknown correspondences, and the solution to a problem equivalent to (14) appears in a single iteration of SICP. We state the results here, and refer the reader to [4] for details.

The optimal $\mathbf{R}$ can be found by computing the matrix $\mathbf{H}$ and taking its SVD,

$$\mathbf{H} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{S}\mathbf{p}'_i\mathbf{q}_i'^T \;\Rightarrow\; \mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V} \tag{15}$$

whereby the rotation matrix is $\mathbf{R} = \mathbf{V}\mathbf{U}^\mathbf{T}$ if $\det(\mathbf{V}\mathbf{U}^T) = +1$, or $\mathbf{R} = \mathbf{V}\begin{pmatrix} \mathbf{I}_{m-1} & 0 \\ 0 & -1 \end{pmatrix}\mathbf{U}^T$ if $\det(\mathbf{V}\mathbf{U}^T) = -1$.

The minimum of (14) with respect to $\mathbf{S}$ can be found by setting the derivative of $F(\mathbf{S}, \mathbf{R})$ with respect to $\mathbf{S}$ equal to zero and solving for the s_j's. This gives the optimal scale parameters,

$$s_j = \underset{s\in[s_{min},s_{max}]}{\operatorname{argmin}} \left| s - \frac{\sum_{i=1}^{n} \mathbf{q}_i'^T\mathbf{R}\mathbf{E}_j\mathbf{p}'_i}{\sum_{i=1}^{n} \mathbf{p}_i'^T\mathbf{E}_j\mathbf{p}_i} \right|, \tag{16}$$

where $\mathbf{E}_j = \operatorname{diag}(0, \ldots, 0, 1, 0, \ldots, 0)$, is one at the j^{th} element, and zero elsewhere, and $[s_{min}, s_{max}]$ are the bounds on admissible values of s. In all the experiments reported below we set $s_{min} = 0, s_{max} = \inf$.

4 Automatic initialization

All of the above methods with the exception of DLT-Fact critically depend on initialization of $\mathbf{t}$, $\mathbf{R}$ and, when input objects are anisotropically scaled, of $\mathbf{S}$. Translation $\mathbf{t}$ is the least problematic of these, as it can be robustly initialized from the observed locations of the image points. But we demonstrate in our experiments that random initialization of $\mathbf{R}$ and $\mathbf{S}$ leads to poor performance. We now describe our method for automatically initializing these parameters. It is inspired by recent successes of example-based methods in vision, including problems of rigid and articulated pose estimation [11, 14].

To achieve this, we propose constructing a database of synthetically rendered projections of objects in the class, with varying aspects and under a range of transformations. We can easily do that by sampling the values of $\mathbf{S}$, $\mathbf{R}$ and $\mathbf{t}$ from suitable distributions. Figure 3 shows a few examples of instances in such a database for box and car classes.

Each instance in the database lists a set of image features: line segments that are projections of 3D model edges, and points that are projections of 3D model vertices. We will refer to these as instance features. Each instance is also associated with the known values of $\mathbf{S}$, $\mathbf{R}$ and $\mathbf{t}$ used to generate it. We will call these the instance parameters. The basic idea behind our initialization method is to look up instances in the database whose features are similar to those in the input image, and to use those instances' parameters to initialize pose estimation. This is illustrated in Figure 1.

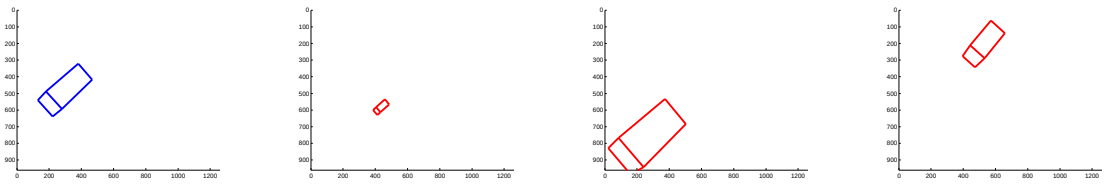


Figure 1: Example of query (left) and three nearest neighbors. We initialize the estimate of $\mathbf{R}$ and $\mathbf{S}$ for the query with the values for the neighbors.

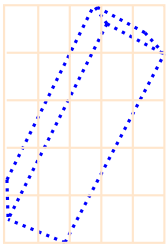


Figure 2: Computing histogram-based descriptor for a set of image features. Points are sampled along line segments corresponding to visible model edges, and counted in each cell of the $B \times B$ histogram, B=5. Note that the aspect ratio of the bounding box is determined by the image features.

Descriptor This requires a method for identifying similar instances. We convert the image features into a sample of image points, as follows. First, all the points given as projections of object vertices are included. Next, for the synthetic database instances, we sample P points for each visible edge (spacing them equally along the segment which is the projection of the edge). If only part of the edge is visible (which can happen in non-convex objects like our car model) we only include in the sample the points along the projection of the edge that are in fact visible. Similarly, for an input image in which we are given a set of line segments, we sample P points along each provided segment.

Given this sample of points, we place a tight bounding box around it, partitioned into a grid of $B \times B$ cells of equal size. Sample points are counted within each cell, as illustrated in Figure 4. The descriptor is then obtained by smoothing the resulting histogram with addition of a uniform pseudocount.

Instance similarity A natural way to look up instances similar to the input object now is to compute the descriptor for the input features, and match them using a χ^2 distance to the descriptors in the database. One issue remains to be resolved, however: Under perspective projection, instances with very similar descriptors appearing in different locations in the image may have significantly different parameters. To account for this we define the following measure of similarity. Let $\mathbf{h}'$ and $\mathbf{h}$ be the two B^2-dimensional histograms, computed for two sets of images features. Let $\mathbf{c}'$ and $\mathbf{c}$ be the centroids of the corresponding point samples. Then, we compute similarity between the two sets as

$$K(\mathbf{h}', \mathbf{h}) \;=\; e^{\chi^2(\mathbf{h}', \mathbf{h}) + \frac{1}{\mu}\|\mathbf{c}' - \mathbf{c}\|^2} \tag{17}$$

The non-negative parameter μ determines sensitivity to location.

Given a test instance with descriptor $\mathbf{h}_0$, we retrieve the top k neighbors - that is, database instances with the highest values of $K(\mathbf{h}_0, \mathbf{h})$. Each of these provides a hypothesis for initialization of $\mathbf{S}$ and $\mathbf{R}$. In all the experiments with iterative methods, we used *each* of the k neighbors as an initial guess, and selected among up to k resulting estimates[1] the best, based on the 2D reprojection error described in 5.3. Note that the latter can be evaluated on a test case without the knowledge of ground truth.

We note that while in our experience the multiple initialization approach described above is sufficiently fast, one can consolidate the information provided by the top k neighbors by clustering them based on 3D error (see Sec. 5.3), and representing each cluster by a single example (and thus a single set of initial pose parameters). This typically results in a reduction from k to about $k/3$ initializations.

Correspondences Until now we have assumed that 2D↔3D correspondences are available to the pose estimation algorithms. But how does one obtain these? In a synthetic experiment we could provide these based on the ground truth known for the test image, but that would be unrealistic. We therefore resort to the following automatic procedure: given one of the nearest neighbors retrieved from the database, correspondences are established by finding the assignment minimizing 2D distances, after the two sets of 2D points are aligned by translation, and scaled to have equal mean

[1] The number could be below k if runs initialized from some of the neighbors fail to converge.

distance from the centroid. If the number of points in both sets is small (e.g. with the box class, which has at most seven visible vertices) we do this exhaustively. Otherwise (e.g. with the car class, with typicaly many more visible vertices), we use a simple greedy assignment algorithm.

5 Experiments

Experiments reported here pursue two goals. One is to evaluate the relative performance of four algorithms: SEO, OIA, SOFTPosit, and the DLT-Fact, under various conditions. The other is to establish the impact of the automatic initialization method proposed above.

5.1 Experimental setup

We are not aware of a database of real images which include detailed annotations for pose and shape of anisotropically scaled object class along with identified image features. Therefore, we built a data set of synthetic images of objects in two classes: boxes and cars. A few examples of each are shown in Figure 3. The canonical model of a box is a cube, defined by 8 vertices, 12 edges, and 6 faces. For cars, we set up an idealized model of a station-wagon type car, containing 14 vertices, 20 edges, and 9 faces. The canonical aspect is 1(length):0.5(width):0.3(height).

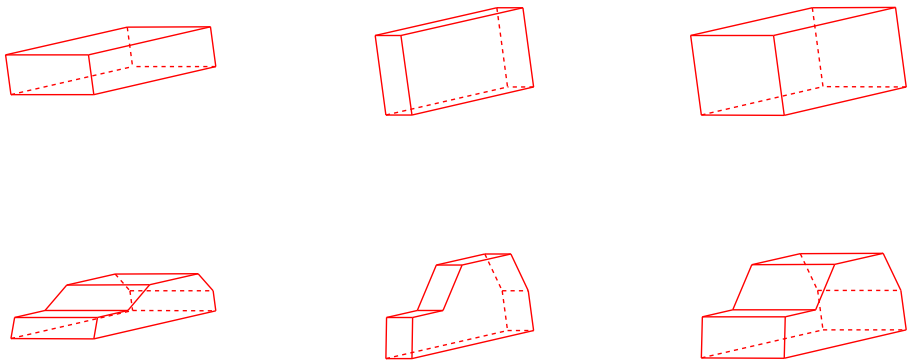


Figure 3: Example instances of box (top) and car (bottom) classes, These share the same rotation, but differ in aspect. Dashed segments are invisible, and would not participate in estimation.

For each of the classes, we created a database of 100,000 instances, computed descriptors and built a database as described in Section 4. We then randomly sampled 1,000 instances to be used as test. For each of the test instances, we remove it from the database prior to testing on it (effectively meaning the database has 99,999 instances).

5.2 Evaluation parameters

There are two axes along which we conducted systematic evaluation: amount of noise and initialization method.

Noise We shift the location of each 2D projection of a visible object vertex independently by adding to it a 2D Gaussian shift vector, drawn from zero-mean Gaussian with covariance $\sigma^2\mathbf{I}_2$. The values of σ are chosen for each vertex as a fixed percentage of the mean length of line segments among the image features that are incident on that vertex. We used $\sigma = 0$ (no noise), 0.01, 0.05, 0.1 and 0.25. Note that with $\sigma = 0.25$ the objects become very significantly distorted.

Initialization We considered the following initialization methods for $\mathbf{R}$ and $\mathbf{S}$ in all the iterative algorithms.

Random Both $\mathbf{S}$ and $\mathbf{R}$ are initialized by random diagonal matrix and random rotation matrix, respectively.

initNN Both $\mathbf{S}$ and $\mathbf{R}$ are set to the values associated with one of the k nearest neighbors found as described in Section 4.

initR $\mathbf{R}$ was initialized from the neighbors, $\mathbf{S}$ is random.

initS $\mathbf{S}$ was initialized from the neighbors, $\mathbf{R}$ is random.

Translation was in all cases initialized according to (7) or (12).

The DLT-Fact algorithm is treated slightly differently from the rest. In the form described in 3.1 it does not require initialization, however it does require correspondences. We establish these correspondences from the top neighbors, and refer to the results under the initNN rubric.

5.3 Error measures

There are three error measures we calculated:

2D reprojection We reproject the canonical model under the transformation defined by the estimated $\mathbf{R}$, $\mathbf{t}$ and $\mathbf{S}$, and compute the mean distances between the reprojected points and the corresponding input vertices. Correspondence is established by a greedy algorithm. The units of this error are essentially pixels.

3D error To estimate rotation, we compare the canonical model rotated by the estimated $\mathbf{R}$, and under the true $\mathbf{R}$, with no translation or scaling, and compute the sum of distances in 3D space between pairs of optimally correspondinc vertices. The units depend on the arbitrary scale of the canonical model, and only provide information on the relative performance of different algorithms.

Aspect error Given the true $\mathbf{S}$ and the estimated $\widehat{\mathbf{S}}$ we normalize the three diagonal values so that the highest is 1, sort them, and compute the Euclidean norm of the difference between the two 3-dimensional vectors. This corresponds to the largest deviation of scaling factors under the best transformation.

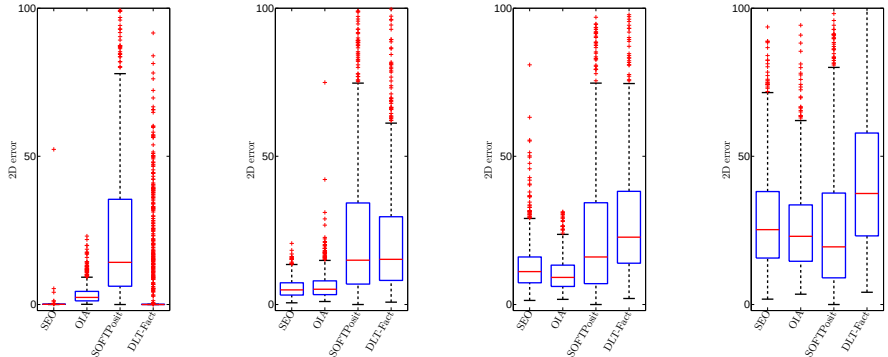
5.4 Results

First we consider the reprojection error. The first set of results, shown in Figure 4, describes the behavior of the four methods intialized with the best behaving method which in all cases is the proposed method initNN. SEO is significantly better under no noise[2], but as noise level increases, the unscaled OIA reduces the gap an eventually outperforms it for high levels of noise (although the differences are not significant). This is due to the fact that under significant noise, the flexibility of SEO becomes a liability. SOFTPosit[3] is the least accurate, however its performance is very stable across noise levels; we presume that the statistical nature of the EM algorithm which makes it relatively robust to noise. DLT-Fact performs well with no noise (note however many outliers), but deteriorates

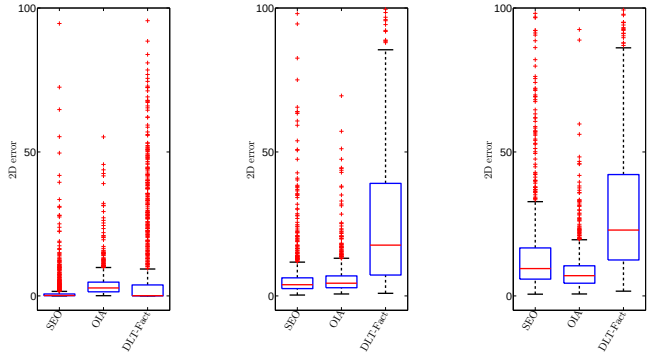
[2] Results under very low noise with $\sigma = 0.01$ (not shown) are qualitatively very to those under $\sigma = 0$.

[3] We could not obtain good convergence behavior with SOFTPosit on the car data, possibly because of the software limitations, and so omit the results here.

as noise increases. We believe this is due to the increasing inaccuracy of QR decomposition that provides the initial estimate for $\mathbf{R}$ and $\mathbf{S}$.



(a) No noise (b) $\sigma = 0.05$ (c) $\sigma = 0.1$ (d) $\sigma = 0.25$



(e) No noise (f) $\sigma = 0.05$ (g) $\sigma = 0.1$

Figure 4: 2D reprojection errors on **boxes** (top) and **cars** (bottom), across noise levels, with initNN. Some of the range is cut off for clarity.

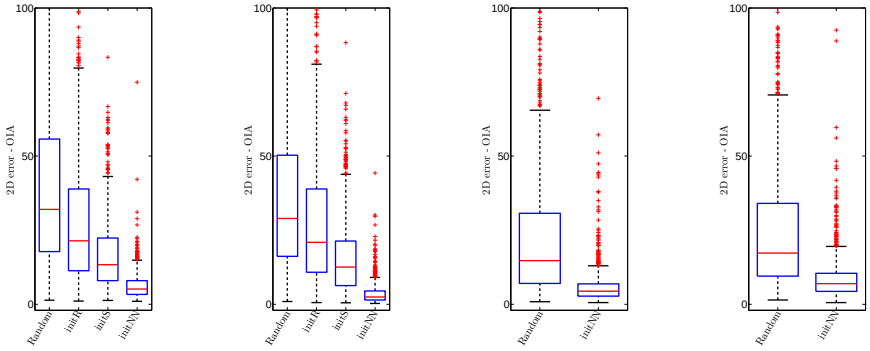
The noiseless or very low noise scenario is interesting since it provides some information on limitations of different methods, and since in some cases it may be realistic. High noise level of $\sigma = 0.25$ is potentially too hard - anecdotally, many objects distorted with this noise are incomprehensible to the human viewer. Thus, in the remainder we focus on the moderate levels of noise, $\sigma \in \{0.05, 0.1\}$.

Figure 6 provides some intuition for interpretation of the 2D error results. Here we show an example for each of the two object classes, in which the 2D error (noted in the figure) is approximately the median for the SEO algorithm under $\sigma = 0.05$ noise.

How much does our automatic initialization improve performance? Figure 5 provides some insight, and shows that across the board, results obtained with initNN are superior.

We can look further into the errors by analyzing separately the 3D error, related purely to rotation, and the aspect error, related to estimated $\mathbf{S}$. Figures 7 and 8 show these quantities, respectively. The relative behavior of the algorithms mirrors that observed for the 2D error.

Absent a database of real images with annotated ground truth, we assess the algorithm's performance on real



(a) $\sigma = 0.05$ (b) $\sigma = 0.1$ (c) $\sigma = 0.05$ (d) $\sigma = 0.1$

Figure 5: Effect of initialization of OIA on **boxes** (top) and **car** (bottom). Some of the range is cut off for clarity.

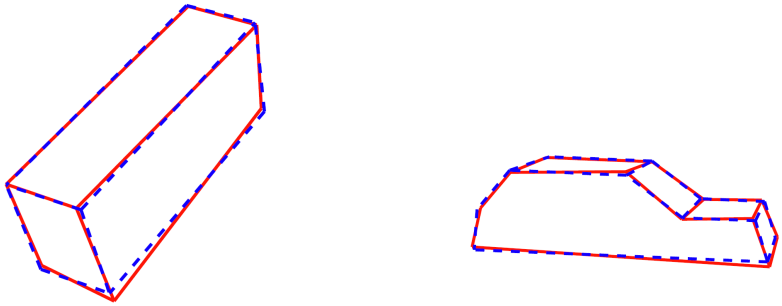
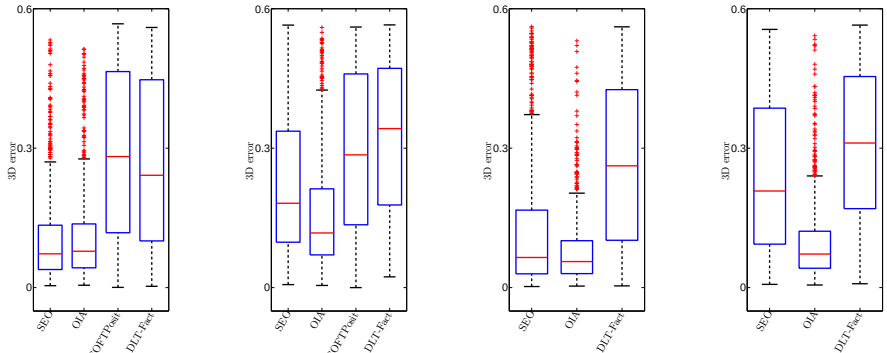
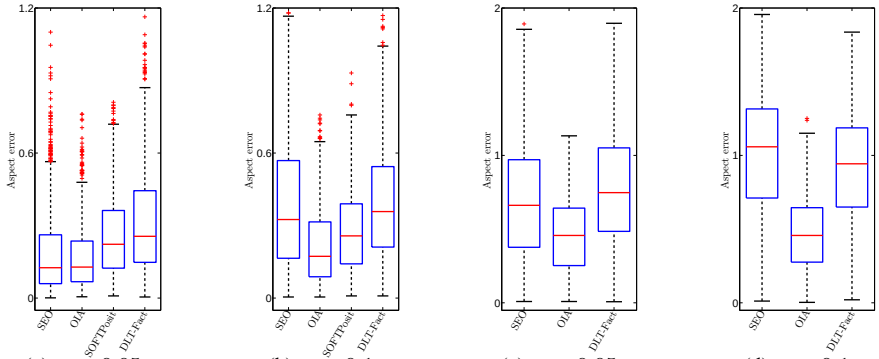


Figure 6: Examples of estimation results by SEO in which 2D error is near the median of plot in Figure **??**, under $\sigma = 0.05$ noise. Solid red: input, dashed blue (nearly overlapping red): estimate.



(a) $\sigma = 0.05$ (b) $\sigma = 0.1$ (c) $\sigma = 0.05$ (d) $\sigma = 0.1$

Figure 7: 3D error on **boxes** (top) and **car** (bottom), measuring accuracy of rotation estimates



(a) $\sigma = 0.05$ (b) $\sigma = 0.1$ (c) $\sigma = 0.05$ (d) $\sigma = 0.1$

Figure 8: Aspect error on **boxes**, measuring accuracy of aspect estimates

Figure 9: Pose estimation for cars in real images. Red: input points, green: projection of the model under estimated $\mathbf{R}$, $\mathbf{t}$ and $\mathbf{S}$.

images anecdotally. Figure 9 shows examples in which we manually marked visible vertices (without specifying correspondences to model). This is of course an approximation, since the actual shape of the cars here is not perfectly fit by our primitive car model. The figure shows the input points, and the estimate obtained by SEO with automatic initialization using the synthetic car database.

6 Conclusion

In this paper we have shown that although well known algorithms for pose estimation are not equipped to deal with anisotropic scaling, they can be extended to this scenario. We also propose an automatic example-based initialization scheme. In our experiments it benefits iterative pose estimation across the board, and we recommend its use instead of random, or weak-perspective based inititalization.

It appears that, given good initialization obtained by our method, no single algorithm is superior to others across noise conditions. Under low noise, both SEO and DLT-Fact produce comparable results superior to those of OIA and especially of SOFTPosit. When noise becomes more severe, DLT-Fact deteriorates significantly, while SEO and OIA do so more gracefully. SEO and DLT-Fact are able to refine the initial estimate of aspect, while the other two methods are not. All in all, our conclusion is to recommend using SEO, especially if estimating aspect is important.

Future work We would like to extend the methods discussed here to more general deformation models, beyond anisotropic scaling, by introducing a statistical model that defines probability distributions for relative locations of vertices or edges. We expect that this will extend the applicability of the approach, and even make it more robust in the anisotropic scaling scenario under noise.

References

[1] M. Arie-Nachimson and R. Basri. Constructing implicit 3d shape models for pose estimation. In *Proceedings of ICCV*, 2010. 1

[2] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on PAMI*, 14(2), 1992. 4

[3] D. DeMenthon and L. S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2), 1995. 3

[4] S. Du, N. Zheng, L. Xiong, S. Ying, and J. Xue. Scaling iterative closest point algorithm for registration of m-d point sets. *Journal of Visual Communication and Image Representation*, 21(5-6), 2010. 4

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2, 4

[6] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA*, 4(4), 1987. 3

[7] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA*, 5(7), 1988. 3

[8] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3), 1987. 2

[9] C. P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on PAMI*, 22, 2000. 2, 3, 4

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2002. 4

[11] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proceedings of ICCV*, 2003. 5

[12] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *Proceedings of British Machine Vision Conference*, 2010. 1

[13] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Proceedings of CVPR*, 2010. 1

[14] G. Taylor, R. Fergus, G. Williams, I. Spiro, and C. Bregler. Pose-sensitive embedding by nonlinear nca regression. In *NIPS*, 2010. 5