Pose Priors for Simultaneously Solving Alignment and Correspondence *

Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua

Computer Vision Laboratory
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

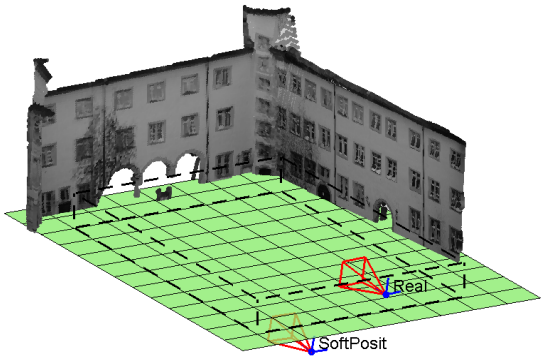
Abstract. Estimating a camera pose given a set of 3D-object and 2D-image feature points is a well understood problem when correspondences are given. However, when such correspondences cannot be established *a priori*, one must simultaneously compute them along with the pose. Most current approaches to solving this problem are too computationally intensive to be practical. An interesting exception is the SoftPosit algorithm, that looks for the solution as the minimum of a suitable objective function. It is arguably one of the best algorithms but its iterative nature means it can fail in the presence of clutter, occlusions, or repetitive patterns. In this paper, we propose an approach that overcomes this limitation by taking advantage of the fact that, in practice, some prior on the camera pose is often available. We model it as a Gaussian Mixture Model that we progressively refine by hypothesizing new correspondences. This rapidly reduces the number of potential matches for each 3D point and lets us explore the pose space more thoroughly than SoftPosit at a similar computational cost. We will demonstrate the superior performance of our approach on both synthetic and real data.

1 Introduction

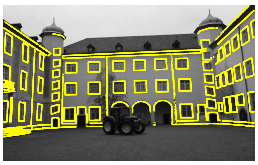
Estimating the pose of a calibrated camera from 3D-to-2D point correspondences between a 3D model and an image is a fundamental problem in Computer Vision. When matches are given, this is known as the Perspective-n-Point (PnP) problem and many effective methods have been proposed. However, as shown in Fig. 1, there are cases where it is difficult to establish the required correspondences, for example because the scene contains repetitive patterns or because the 3D points are simply salient features on a CAD model without associated texture.

In such cases, it becomes necessary to simultaneously compute the pose and establish the correspondences. As shown in Fig. 1, SoftPosit [1], arguably the most computationally effective algorithm able to do this based on geometry alone, tends to fail in presence of large amounts of clutter, occlusions, or repetitive patterns. Other methods can avoid these issues by using a RANSAC-style [2] approach [3, 4] or searching for clusters in the pose space [5–8], but quickly become computationally intractable with realistic numbers of features.

* This work was supported by the Swiss National Science Foundation and by funds of the European Commission under the IST-project 034307 DYVINE.



(a)



(b) Blind PnP

(c) SoftPosit

Fig. 1. Recovering the pose in a scene with repetitive patterns. (a) 3D model of the scene. The "Real" camera, and the pose recovered by SoftPosit [1] are indicated. Blind PnP is able to retrieve the "Real" pose. (b) Model reprojected after estimating the pose using Blind PnP. (c) Model reprojected after estimating the pose using SoftPosit.

In this paper, we propose an approach that is much more robust to occlusion, clutter, and repetitive patterns than SoftPosit without increase in computational complexity. We refer to our algorithm as *Blind PnP* because it solves the PnP problem without being given the correspondences. It relies on the fact that, in practice, prior information on the camera pose is often available. For example, in the case depicted by Fig. 1, the camera must be pointing towards the building and be above ground level. We model such a prior as a Gaussian Mixture Model (GMM) and use each component of the GMM to initialize a Kalman filter. We then explore the space of possible correspondences within a subset of potential matches and keep the hypothesis that yields the smallest reprojection error.

As illustrated by Fig. 2, the Kalman filter guides and speeds up the matching process, while preventing gross errors. As a result, the complexity of our method is of order $\mathcal{O}(Gn^3M \log N)$, where G is the number of components in the GMM, M the number of 3D points, N the number of image features, and n the number of 2D correspondences that can be potentially matched to each 3D point when exploiting the prior. This turns out to be comparable to the complexity of SoftPosit, which is $\mathcal{O}(MN^2)$. However, since n is typically much smaller than N, our approach can perform a more exhaustive search for a similar amount of computation and therefore be less sensitive to distractors such as clutter.

Of course, further increases in speed could be achieved by considering image clues. For example, with the repetitive patterns of Fig. 1, it may not be possible assign to a 3D feature a single image feature on the sole basis of local image information but this information can nevertheless be used to restrict the number of potential matches. This is entirely compatible with the framework we propose.

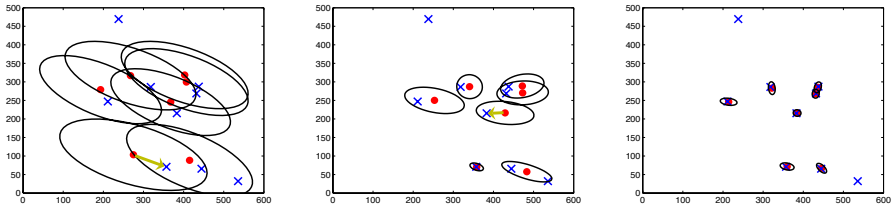


Fig. 2. Blind PnP. **Left:** The dots represent the projected 3D points and the ellipses the corresponding uncertainty regions given the pose prior. The crosses represent the 2D points. Even before hypothesizing any correspondence, the prior considerably reduces the set of potential matches. **Center:** Given a single matching hypothesis represented by the arrow, the number of possibilities for the second match is further reduced. **Right:** This is even truer of the third match and, once it has been hypothesized, the uncertainty ellipses become so small that finding additional matches becomes easy.

2 Related Work

Simultaneously recovering poses and correspondences has traditionally been achieved by hypothesizing small sets of k 3D-to-2D correspondences between the M 3D points and N 2D points, and validating the poses they generate using the remaining points [3]. Using the RANSAC algorithm [2] the computational complexity is $\mathcal{O}(MN^k \log N)$, which is prohibitive when the number features becomes large. This can be partially addressed by means of heuristic criteria to terminate the search as soon as a "good solution" is found [9, 10, 4]. Nevertheless, the algorithms remain exceedingly slow for realistic values of k, M and N.

Another approach to reducing the computational burden is to focus on specific regions of the search space. This can be done in a data-driven manner using indexing methods [11–13]. These techniques initially learn 2D feature groupings of 3D objects and store the associated vectors in data structures, such as Hash tables [11, 12], or kd-trees [13]. At runtime, feature vectors are extracted from the test images and used to access the database to extract the 3D-to-2D hypotheses, which are then used for pose estimation. These techniques, however, can only handle relatively small pose spaces.

Specific subspaces of the search space can also be preselected in a model driven fashion, for example through a pose clustering approach [5–8, 14]. These techniques consider all possible 3D-to-2D matches, and the poses they generate are represented in a 6D pose space. High-probability clusters are then extracted from this space, under the assumption they will contain only correctly hypothesized matches. Finally, the hypotheses within these clusters are further considered for verification and pose estimation. In fact, pose clustering techniques are similar to the hypothesize-and-test algorithms discussed above, in the sense that they initially take into account all possible matches. As a consequence they also lead to computational explosion when considering large sets of 3D-to-2D correspondences. For instance, one of the most efficient algorithms within this group [7], has a complexity of $\mathcal{O}(MN^3)$.

Other approaches iteratively solve for pose and correspondence [1, 15, 16], by optimizing a global cost function. Among these, SoftPosit [1] is the one that stands out because of its accuracy and efficiency. It combines an iterative pose estimation technique, with an algorithm to assign correspondences, resulting in an $\mathcal{O}(MN^2)$ complexity. One limitation of such an algorithm is that the global minimum can not be guaranteed. This is alleviated by randomly initializing it at different initial guesses. Yet, certain configurations of the data or situations with large amounts of occlusion and clutter, still cause SoftPosit to fail.

Our approach differs from the previous ones in that we approximate the 6D pose space as a GMM, meaning that we define several initial pose guesses with an associated uncertainty. Each Gaussian component is independently verified, with the advantage that only a very small subset of 3D-to-2D correspondences needs to be hypothesized. In contrast, the "hypothesis-and-test" and "pose clustering" approaches consider all possible hypotheses. On the other hand, the iterative algorithms, although started from many initial guesses, do not exploit the fact that given one such guess, the set of plausible 3D-to-2D correspondences is much reduced. These key differences allow Blind PnP to be as efficient as the SoftPosit algorithm while retaining the robustness of exhaustive techniques.

3 Pose Priors for Alignment and Correspondence

As shown in Fig. 2, we use the pose priors to restrict the number of possible 3D-to-2D matches. Hypothesizing one such match decreases the pose uncertainty, further reducing the possibilities for the second match, and even more so for the third. Once three correspondences have been hypothesized, the uncertainty becomes so small that we can estimate a pose and match other 3D points to 2D points that are close enough to their projection using that estimate. We can then evaluate its quality using a suitable objective function, iterate over all plausible triplets of correspondences, and retain the best one.

3.1 Formalization

We assume we are given a set $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ of M 3D model points and a set $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ of N 2D image points, coming for instance from a keypoint detector. Some of the 2D points correspond to 3D points and some do not. Similarly, the projections of some of the 3D points may not be among the 2D points. We suppose as well we are given a set of pose prior samples. Using Expectation Maximization we represent them as a Gaussian Mixture Model that includes G Gaussian components with 6-vectors $\{\mathbf{p}_1, \ldots, \mathbf{p}_G\}$ for the means, and 6×6 covariance matrices $\{\mathbf{\Sigma}_1^p, \ldots, \mathbf{\Sigma}_G^p\}$ [1]. Fig. 3 shows an example of such pose priors.

Our goal is to find both the correct pose $\mathbf{p}$ and as many 2D-to-3D correspondences as possible. Let Matches be a set of $(\mathbf{x}, \mathbf{u})$ pairs that represents those

[1] We use exponential maps to parametrize the pose rotation, yielding to 6 values to parameterize the complete pose.

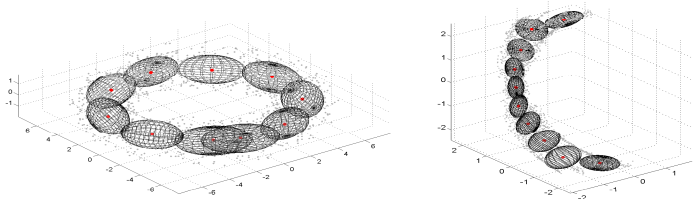


Fig. 3. Example of pose samples and the corresponding Gaussian components for a case where the pose is constrained to be inside a torus surrounding the object and pointing to it. For simplicity, we represent the translation (left) and rotation (right) spaces separately and do not represent the cross-covariances between the two spaces.

correspondences and NotDetected be the subset of 3D points for which no match can be established. We want to find the correct $\mathbf{p}$ and Matches by minimizing

$$\text{Error}(\mathbf{p}) = \sum_{(\mathbf{x},\mathbf{u})\in\text{Matches}} \|\mathbf{u} - \text{Proj}(\mathbf{p};\mathbf{x})\| + \left|\text{NotDetected}\right|\mathcal{T}\,, \tag{1}$$

where $\text{Proj}(\mathbf{p};\mathbf{x})$ is the 2D projection of the 3D point $\mathbf{x}$ given pose $\mathbf{p}$, and $\mathcal{T}$ is a penalty term that penalizes unmatched points and whose value will be set as discussed below. $\left|\cdot\right|$ accounts for the cardinality of a set.

We next turn to the steps of our approach. The pseudocode is given in Fig. 4.

3.2 Algorithmic Steps

Limiting the possibilities for a first match. We want to minimize (1) using the pose priors. Each Gaussian component gives an initial $\mathbf{p}_g$ pose estimate and an associated covariance $\mathbf{\Sigma}_g^p$. We use this covariance to restrict the possibility of matches between the 3D and 2D points, by computing the projections $\mathbf{v}_i$ and the covariances $\mathbf{\Sigma}_i^v$ of the $\mathbf{x}_i$ 3D points. Error propagation yields

$$\mathbf{\Sigma}_i^v = \mathbf{J}(\mathbf{x}_i)\ \mathbf{\Sigma}_g^p\ \left(\mathbf{J}(\mathbf{x}_i)\right)^\top\,, \tag{2}$$

where $\mathbf{J}(\mathbf{x}_i)$ is the Jacobian of $\text{Proj}(.;\mathbf{x}_i)$. This defines a search region for the point $\mathbf{x}_i$, and we only consider the 2D points $\mathbf{u}_j$ such that

$$(\mathbf{v}_i - \mathbf{u}_j)^\top\ \mathbf{\Sigma}_i^v\,(\mathbf{v}_i - \mathbf{u}_j) \leq \mathcal{M}^2 \tag{3}$$

as potential matches for $\mathbf{x}_i$. $\mathcal{M}$ is a threshold chosen to achieve a specified degree of confidence, based on the cumulative chi-squared distribution. In our experiments we use $\mathcal{M}$=2 and 3, which yield 86% and 99% of confidence, respectively. This reduces the search space from the entire image to a small elliptical region.

Updating the pose and its uncertainty from the first correspondence. Among 3D points $\mathbf{x}_i$ that have at least one potential match, we begin by those

```
BestSolution ← ⊥
For each Gaussian component g
| BuildMatchesTriplets(p_g, Σ^p_g, ∅, BestSolution)
Best pose and matches are in BestSolution

Function BuildMatchesTriplets(p, Σ^p, Hypotheses, BestSolution)
| If all x_i already appear in Hypotheses
| | Return
| For each x_i that do not appear in Hypotheses
| | v_i ← Proj(p; x_i) ;  Σ^v_i = J(x_i) Σ^p (J(x_i))^⊤
| | S_i ← {u_j | (v_i − u_j)^⊤ Σ^v_i (v_i − u_j) ≤ M^2}
| m ← argmin_i |S_i|
| For each u_j ∈ S_m
| | p^+ ← K(u_j − v_m); Σ^{p+} ← (I − KJ(x_m)) Σ^p
| | If already 2 matches in Hypotheses
| | | Find optimal matches using p^+ and Σ^{p+} and put them in OptimalMatches
| | | Compute pose p' and Error from matches in OptimalMatches
| | | If Error < TerminationThreshold
| | | | Early Termination with pose p' and matches in OptimalMatches
| | | If Error < Error( BestSolution )
| | | | BestSolution ← (p', OptimalMatches )
| | | Else
| | | BuildMatchesTriplets(p^+, Σ^{p+}, Hypotheses ⊕ [x_m ↔ u_j], BestSolution)
| If number of consecutive not detected points in Hypotheses < Threshold
| | BuildMatchesTriplets(p, Σ^p, Hypotheses ⊕ [x_m is not detected], BestSolution)
End function
```

Fig. 4. Pseudo-code of the Blind PnP.

that have the fewest and consider each one in turn. Hypothesizing a correspondence between $\mathbf{x}_i$ and $\mathbf{u}_j$ reduces the pose uncertainty. We use standard Kalman filter equations to update the $\mathbf{p}_g$ pose estimate and its covariance $\mathbf{\Sigma}_g$ and write

$$\mathbf{p}_g^+ = \mathbf{p}_g + \mathbf{K}\left(\mathbf{u}_j - \text{Proj}(\mathbf{p}_g; \mathbf{x}_i)\right) \ , \tag{4}$$

$$\mathbf{\Sigma}_g^{p+} = \left(\mathbf{I} - \mathbf{K}\mathbf{J}(\mathbf{x}_i)\right)\mathbf{\Sigma}_g^p \ , \tag{5}$$

where $\mathbf{K}$ is the Kalman Gain and $\mathbf{I}$ the Identity matrix.

Building sets of three correspondences. After this update, the pose covariance is much smaller and the number of potential hypotheses for the second match will be highly reduced. The process is repeated one last time to select a third hypothesis. In general, the innovation term in (4) then becomes almost negligible. This is consistent with the fact that only three 3D-to-2D matches are needed to compute the camera pose.

Finding optimal correspondences. Therefore, after three match hypotheses, the pose does not evolve anymore and we can easily match the remaining 3D points by projecting them onto the image and assigning to them the closest of the N feature points. This assignment can be done in $\mathcal{O}(\log N)$ [17] for each point. The 3D points projected farther apart from any 2D feature than a fixed distance $\mathcal{T}$ are considered as not detected. Following a similar discussion as for $\mathcal{M}$ in (3), $\mathcal{T}$ is set to $9\sigma^2$, with σ^2 being the variance of the Normal image noise. This ensures with 99% probability that a 2D feature and a 3D projected point actually match.

Iterating and terminating. We repeat the process above by iterating over the Gaussian components, and the 3D points $\mathbf{x}_i$ with their constrained set of potential 2D correspondents given by (3). We stop if the error in (1) drops below a threshold, taken sufficiently small to reduce the risks of false positives.

Dealing with non-detected 3D points. We next explain how to deal with 3D points lacking of a 2D match. Once a pose has been computed from three correspondences, we handle this possibility by considering as not detected the 3D points projected farther apart from any 2D image feature than a fixed distance $\mathcal{T}$. However this is not sufficient because the points in the initial sets of three matches may have also not been detected. After iterating over the potential 2D correspondents $\mathbf{u}_j$ for $\mathbf{x}_i$, we consider the case that $\mathbf{x}_i$ may not have been detected, skip it, and consider the next best 3D point for the next correspondences.

But postponing the making of hypotheses may increase dramatically the algorithm complexity. Fortunately this can be avoided because the probability to consecutively pick r non-detected 3D points quickly decreases when r increases. This probability can be modeled as a sampling without replacement process:

$$P(\mathbf{x}_{i_1} \in \mathsf{NotDetected}, \ldots, \mathbf{x}_{i_r} \in \mathsf{NotDetected}) = \frac{N_{\mathrm{nd}}!\ (M-r)!}{(N_{\mathrm{nd}}-r)!\ M!}\,, \qquad (6)$$

where $N_{\mathrm{nd}} = \big|\mathsf{NotDetected}\big|$ is the number of non-detected 3D points. When this probability drops below a fixed threshold —we use 5% in all our experiments—, we stop considering the 3D points as non-detected. Note that the number of non-detected N_{nd} is not known *a priori*. However, conservatively setting this number to a sufficiently large and fixed value —we assume a ratio of 60% of outliers in all our experiments— prevents erroneously stopping the search when the actual number of non detected points is smaller.

4 Complexity Analysis

We now describe the complexity of our algorithm. Our goal is to estimate the number R of 3D-to-2D hypotheses that must be drawn in order to ensure with probability λ that at least one of them is outlier-free.

Estimating the pose requires three correct correspondences, which are both detected and correctly assigned to a 2D feature. The probability of this to happen for three randomly chosen model points is:

$$P_{\mathsf{3detected}} = \frac{M_d}{M} \cdot \frac{M_d-1}{M-1} \cdot \frac{M_d-2}{M-2} \qquad P_{\mathsf{3correct\ assign}} = \frac{1}{n_1\ n_2\ n_3} \approx \frac{1}{n^3}$$

where $M_d = p_d M$ stands for the number of detected model points, p_d is the ratio of detected model points, n_1 is the number of 2D points in the first uncertainty region, n_2 is the number of 2D points in the uncertainty region after the first hypothesis, and n_3 the same but after the second hypothesis. Since the uncertainty regions progressively shrink, we have $n_1 > n_2 > n_3$, but for simplicity we write the right-most approximation, where n is usually much smaller than the total

number N of 2D points. From the previous equations we get the probability that a set of three matches has only inlier correspondences:

$$P_{\text{3correct matches}} = P_{\text{3detected}} \times P_{\text{3correct assign}} \approx \frac{M_d^3}{M^3 n^3} = \left(\frac{p_d}{n}\right)^3 . \quad (7)$$

We can then write the probability of taking R consecutive incorrect samples as $(1 - P_{\text{3correct matches}})^R$. Hence, the number of samples R that need to be drawn to ensure with probability λ that at least one of them is correct is such that $(1 - P_{\text{3correct matches}})^R \leq \lambda$, which leads[2] to

$$R \approx \left(\frac{n}{p_d}\right)^3 \log\left(\frac{1}{1-\lambda}\right) . \quad (8)$$

Each time 3 model points are matched to 3 image features, the pose is recomputed in constant time using a Kalman Filter. All the model points are then projected onto the image in time $\mathcal{O}(M)$ and the correctness of the pose is verified by assigning image points to the projected model points. The assignment is done in time $\mathcal{O}(\log N)$ following [17]. This process has to be repeated in the worst case for all the G Gaussian components. Thus, the final cost of our algorithm is:

$$\left(\frac{n}{p_d}\right)^3 \log\left(\frac{1}{1-\lambda}\right) \times \mathcal{O}(G) \times \mathcal{O}(M) \times \mathcal{O}(\log N) = \mathcal{O}(Gn^3 M \log N) \quad (9)$$

where G and n are much smaller than N. Observe that with the probabilistic partition of the pose space, the complexity of our algorithm is significantly smaller to the $\mathcal{O}(MN^3 \log N)$ complexity of the general hypothesize-and-test approach [2, 4], or the $\mathcal{O}(MN^3)$ of the traditional pose clustering algorithms [7] which still consider all the possible 3D-to-2D correspondences.

The computational cost of our approach is comparable to the $\mathcal{O}(MN^2)$ complexity of SoftPosit. Consider for instance a case of a 3D model with M=80 and N=100. Typical values for G and n are 20 and 5, respectively. This would mean that a hypothesize-and-test approach would require 3.6×10^8 operations to solve the problem, a pose clustering would require 8.0×10^7 operations, while SoftPosit and Blind PnP would require 8.0×10^5 and 9.2×10^5 operations, respectively.

From (9), it can be seen that the best efficiency of the Blind PnP is achieved when image points are more or less evenly spread over the image. In this situation, the number n of image points falling into each ellipse of uncertainty tends to be small, and so is the complexity of the algorithm. In contrast, a hypothetical image where all the points were concentrated in a small region, would be the worst situation for our algorithm —its performance would converge to that of the hypothesize-and-test algorithms. Nevertheless, this situation is both unusual, and undesirable for any pose estimation algorithms because it would lead to many ambiguities in the 3D-to-2D correspondences.

[2] Since $(p_d/n)^3 << 1$ we use the approximation $\log(1-x) \approx -x$ for small x.

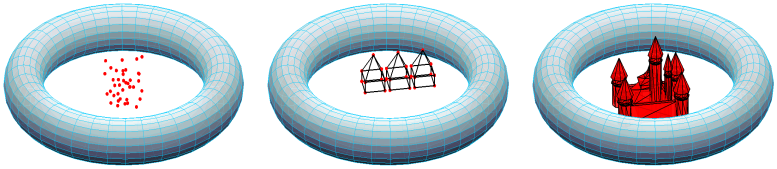


Fig. 5. 3D point configurations and possible locations for the camera centers. **Left:** From 20 to 80 randomly distributed points. **Middle:** 27 points forming repetitive structures. **Right:** 100 points laying on the CAD model of a castle.

5 Results

We evaluated our approach on both synthetic and real data by comparing it to SoftPosit. To ensure fairness, we initialized SoftPosit with the means of the GMM components we use to represent pose priors, instead of randomly. Since there is no obvious way to exploit the pose covariances in SoftPosit, we ran it several times using different values of a parameter β that controls the uncertainty of the initial pose [1]. To eliminate spurious local minima, we used the same stopping criterion for SoftPosit as for Blind PnP, which is to stop the search only if the residual in Eq. 1 drops below a very small threshold, which is much stricter than the one in [1]. Therefore, we compare our method to a Softposit version that was enhanced in order to take advantage of the pose priors.

To estimate convergence rates given the true camera rotation $\mathbf{R}_{true}$ and translation $\mathbf{t}_{true}$, we first computed the relative error of the estimated rotation $\mathbf{R}$ by $E_{rot}(\%) = \|\mathbf{q}_{true} - \mathbf{q}\|/\|\mathbf{q}\|$, where $\mathbf{q}$ and $\mathbf{q}_{true}$ are the normalized quaternions corresponding to the rotation matrices. Similarly, the relative error of the estimated translation $\mathbf{t}$ was taken to be $E_{trans}(\%) = \|\mathbf{t}_{true} - \mathbf{t}\|/\|\mathbf{t}\|$. Finally, we considered a pose to be correct if both $E_{rot}(\%)$ and $E_{trans}(\%)$ were below 10%.

5.1 Synthetic Experiments

We peformed numerous MonteCarlo simulations to account for clutter, occlusions and different 3D point configurations. Following [1], each simulation was characterized by the number M of 3D model points, the fraction $p_d \in \{0.4, 0.6, 0.8\}$ of occluded 3D points with no corresponding 2D point, and the percentage $p_c \in \{0.6, 0.8\}$ of clutter, that is, 2D points with no corresponding 3D point. For each set of parameter values, we performed $n_t = 50$ random and independent trials. For each trial, we ran SoftPosit three times with different values of $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}\}$. The Blind PnP results of this section were obtained with the Mahalanobis boundary $\mathcal{M}$ on (3) set to 2. For comparison purposes, we also ran our algorithm with $\mathcal{M}=3$ and obtained virtually indistinguishable convergence rates but an increase in computation time by a factor of 2 to 3.

Fig. 5 depicts the three kinds of 3D points configurations—randomly distributed, repetitive structures, and lying on a CAD model of a castle—along with the regions of possible camera locations we used. The camera optical axes

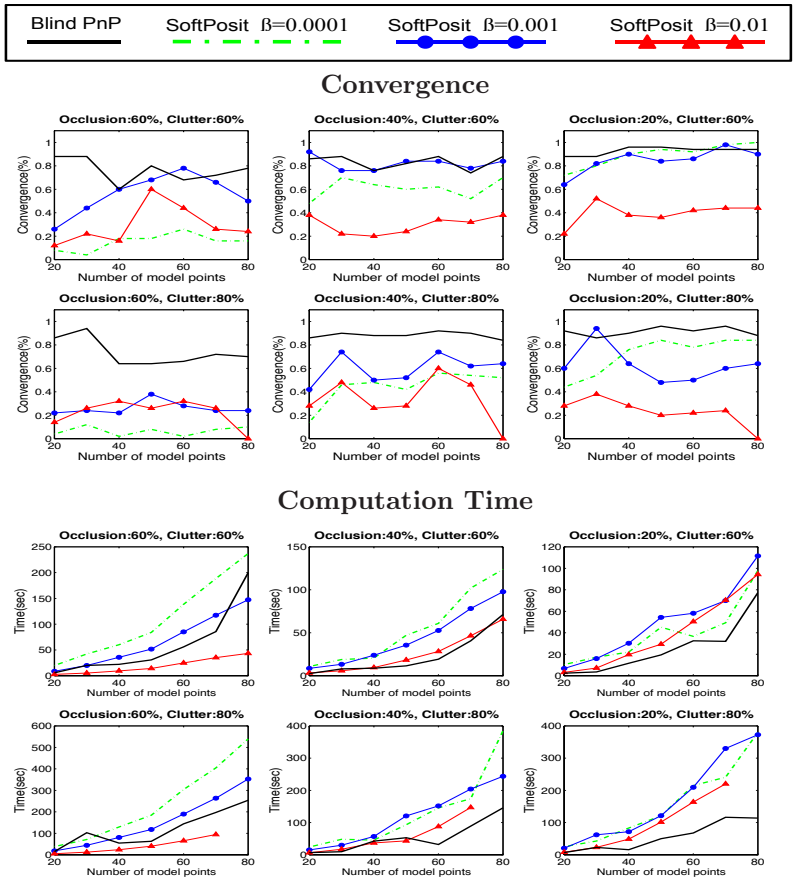


Fig. 6. Synthetic experiment using the randomly distributed points of Fig. 5. Top row: Convergence rate. Bottom row: Computation time per trial.

were allowed to point anywhere on the 3D model. The complete 6D space of possible poses was approximated by a 20-component GMM. The 2D points were produced by generating random poses within the 6D space and projecting the model onto a 640×480 image using a virtual calibrated camera with an effective focal length of f=800, and a principal point at (u_c, v_c)=(320, 240). Normal noise with standard deviation σ=2 was then added to the image coordinates of those projections, and clutter points with random coordinates were created.

Fig. 6 depicts the results on the random configuration of 3D points, with $M \in \{20, 30, \ldots, 80\}$. In terms of convergence, for 60% of clutter, Blind PnP and SoftPosit perform similarly. However, when the percentage of clutter increases to 80%, Blind PnP outperforms SoftPosit consistently. Computation times using `MATLAB` implementations of both algorithms are comparable, thus confirming the

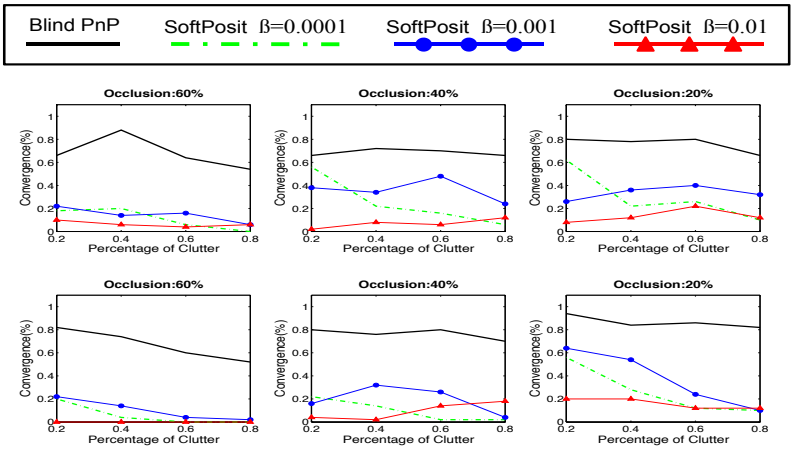


Fig. 7. Convergence rates when using the structured points of Fig. 5. Top row: Points forming repetitive structures. Bottom row: Points laying on the CAD model of a castle.

results of our complexity analysis. The advantage of Blind PnP becomes even more marked when dealing with 3D points that form repetitive structures, such as the two rightmost sets of Fig. 5. As shown in Fig. 7, Blind PnP outperforms SoftPosit independently of the occlusion or clutter levels in both cases.

5.2 Real Data

We used several images of the scenes of Fig. 8 and 10 to validate our approach on real data. Before summarizing the results on these data sets, we first briefly present our approach to extracting 3D model points and 2D image features.

Extracting 3D and 2D Feature Points. A common approach to selecting 3D features is to pick corners and line intersections in the 3D model [1, 7, 18, 14]. This implies that the object has crisp edges, which may not always be true. We instead manually registered the 3D model to one of the images from which we extracted feature points using the SIFT keypoint detector [19]. We then obtained our 3D points by simply backprojecting these 2D points to the model. A potential problem with this approach is that SIFT can return many thousands keypoints per image, which is far more than we can handle: As shown in Fig. 6 Blind PnP and SoftPosit can handle approximately about M=100 to 150 points in a reasonable amount of time, which is already more than earlier approaches.

We reduced the number of detected features using the simple clustering approach depicted by Fig. 8. The initially numerous keypoints were grouped into a few clusters using a k-means algorithm based on proximity. Then, within each cluster, a percentage of the keypoints with the largest gradient were selected. We also ran this simple procedure to extract 2D features from the images on which we tested our algorithm and found that it consistently reduced the number of keypoints from a few thousands to about a hundred in a stable way.

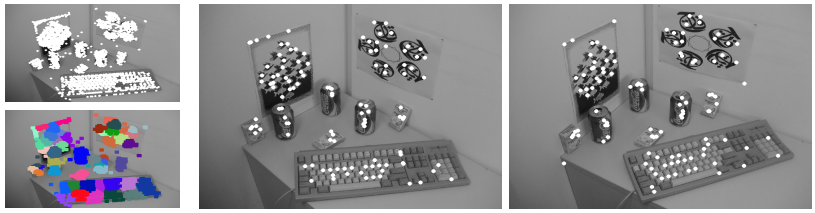


Fig. 8. Feature extraction. **Left:** A few thousand keypoints extracted from the image are clustered using a k-means algorithm. **Middle:** Only the most salient keypoints in each cluster are retained. **Right:** The same procedure is run on a different image and yields many of the same keypoints.

Occluded Indoor Scene. For the scene of Fig. 8, we acquired 17 different images by following a roughly circular path. We created the 3D model shown as a wireframe in Fig. 9 and registered it to one these images using Image Modeler (©2005 Realviz s.a.). The method described above produced M=100 3D points in that image and N between 110 and 125 2D points in all others. The camera pose prior was modeled by a $G=20$ component GMM. The camera locations were assumed to be within a quarter-hemisphere, bounded by the two walls and the table while the optical axis was allowed to point anywhere on the 3D model.

To validate the algorithms under occlusion, we ran many trials for each image by synthetically removing all the keypoints falling within parts of the images, such as the black triangles of Fig. 9. As before, we ran SoftPosit using $\beta=\{10^{-4},10^{-3},10^{-2}\}$ and retained the best result. Similarly, to improve the convergence rate of Blind PnP on this noisy real-world data, we ran it first with Mahalanobis boundaries $\mathcal{M}=2$ and ran it again with $\mathcal{M}=3$ if it failed to converge. This improved our rates by more than 30% with only a limited computational overhead since the more expensive computations were only performed when absolutely necessary. In the images on the left side of Fig. 9, we reproject the model using the pose recovered by either Blind PnP or SoftPosit. To quantify these results, we also registered manually the 3D model in the test images and computed convergence rates. The graph on the right side of Fig. 9 summarizes these results for all trials on all images and confirms that Blind PnP is much more robust than SoftPosit to increasing occlusion levels.

Repetitive Outdoor Scene. Finally, we come back to the scene of Fig. 1 and its repetitive structure. The data set was composed of 8 images of an inner patio with windows, arches, and other elements evenly distributed over the scene. The 3D model was obtained using a laser range-finder and was manually registered to one of the images in order to obtain M=120 3D points. In the other images, around N=100 2D features were detected. Fig. 1(a) shows the boundaries of the $40\times30\times2$ m^3 volume defining the prior for camera locations. The camera was allowed to point any point of the 3D model. This pose space was approximated by G=$\{20,30,40,60,100\}$ Gaussian components. Fig. 10 shows 3 samples of model

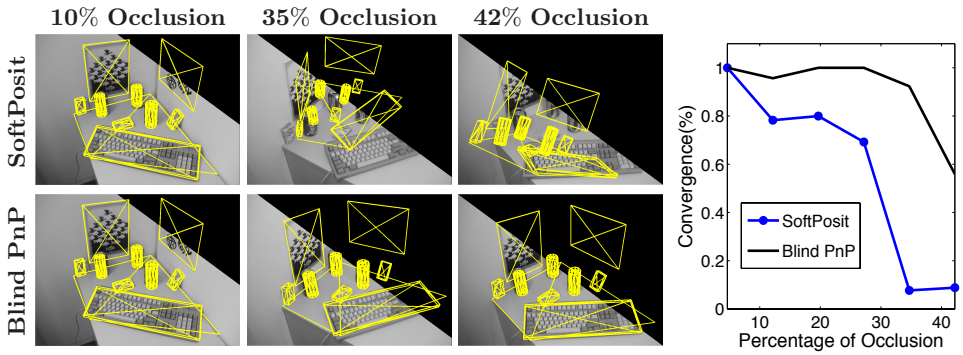


Fig. 9. Indoor results. Upper row images: 3D model reprojected using the SoftPosit pose. Lower row images: Using the Blind PnP pose. Graph: The convergence rate as a function of the percententage of occluded 3D points.

reprojection after recovering the pose using the Blind PnP and SoftPosit. By simply initializing the Blind PnP twice, with $\mathcal{M}=\{2,3\}$, we were able to retrieve the correct camera pose in all 7 test images using only $G=20$ components. By contrast, SoftPosit failed to retrieve correct poses for 5 of that 7 images even when it was initialized up to $G=100$ times, and with $\beta=\{10^{-4}, 10^{-3}, 10^{-2}\}$.

6 Conclusion

We have shown that introducing very weak priors on camera pose leads to efficient, robust, and simultaneous estimation of pose and of 3D-to-2D correspondences. The priors are represented by a GMM whose components we use to initialize a Kalman filter. The filter is used to restrict the set of 2D points that can be associated to the 3D features and its covariance is progressively reduced as successive matches are hypothesized. This gives us robustness to clutter, occlusions, and repetitive patterns. The approach presented here relies solely in geometry. In practice, even when one-to-one correspondences cannot be reliably established between model and image, image clues can be used to restrict the set of possible matches. Furthermore, when dealing with videos, the pose priors we use could be made much stronger based on motion clues. In future work, we will incorporate these additional sources into our algorithm, which should yield significant speed increases and make Blind PnP fit for real-time applications.

References

1. David, P., DeMenthon, D., Duraiswami, R., Samet, H.: Softposit: Simultaneous pose and correspondence determination. IJCV **59**, 259–284 (2004)
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**, 381–395 (1981)

Fig. 10. Experiment 5. Upper row: SoftPosit. Lower row: Blind PnP. The images and 3D model for this experiment are courtesy of PhD Christoff Strecha.

3. Grimson, W.E.L., Meyer-Arendt, J.R.: Object recognition by computer: the role of geometric constraints. MIT Press: Cambridge, MA. **31** (1992)
4. Grimson, W.E.L.: The combinatorics of heuristic search termination for object recognition in cluttered environments. IEEE PAMI. **13**, 920–935 (1991)
5. Breuel, T.: Fast recognition using adaptive subdivisions of transformation space. In: Proc. IEEE CVPR., 445–451 (1992)
6. Cass, T.A.: Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. In: Proc. ECCV., 834–842 (1992)
7. Olson, C.F.: Efficient pose clustering using a randomized algorithm. IJCV **23**, 131–147 (1997)
8. Stockman, G.: Object recognition and localization via pose clustering. Comput. Vision Graph. Image Process. **40**, 361–387 (1987)
9. Ayache, N., Faugeras, O.D.: Hyper: a new approach for the recognition and positioning to two-dimensional objects. IEEE PAMI. **8**, 44–54 (1986)
10. Grimson, W.E.L., Lozano-Pérez, T.: Localizing overlapping parts by searching the interpretation tree. IEEE PAMI. **9**, 469–482 (1987)
11. Lamdan, Y., Wolfson, H.J.: Geometric hashing: A general and efficient model-based recognition scheme. In: Proc. IEEE ICCV., 238–249 (1988)
12. Burns, J.B., Weiss, R.S., Riseman, E.M.: View variation of point-set and line-segment features. IEEE PAMI. **15**, 51–68 (1993)
13. Beis, J.S., Lowe, D.G.: Indexing without invariants in 3D object recognition. IEEE PAMI. **21**, 1000–1015 (1999)
14. Jurie, F.: Solution of the simultaneous pose and correspondence problem using gaussian error model. CVIU. **73**, 357–373 (1999)
15. Wunsch, P., Hirzinger, G.: Registration of cad-models to images by iterative inverse perspective matching. In: Proc. ICPR., 78 (1996)
16. Beveridge, J.R., Riseman, E.M.: Optimal geometric model matching under full 3D perspective. CVIU. **61**, 351–364 (1995)
17. Arya, S., Mount, D., Netanyahu, N., Silverman, R., Wu, A.: An optimal algorithm for approximate NN searching fixed dimensions. J.ACM **45**, 891-923 (1998)
18. David, P., DeMenthon, D.: Object recognition in high clutter images using line features. In: Proc. IEEE ICCV., 1581–1588 (2005)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **15**, 91–110 (2004)