IDIAP RESEARCH REPORT

LETHA: LEARNING FROM HIGH QUALITY INPUTS FOR 3D POSE ESTIMATION IN LOW QUALITY IMAGES.

Adrian Penate-Sanchez Francesc Moreno-Noguer
Juan Andrade-Cetto Francois Fleuret

Idiap-RR-22-2014

DECEMBER 2014

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

LETHA: Learning from High Quality Inputs for 3D Pose Estimation in Low Quality Images

Adrian Penate-Sanchez[1] Francesc Moreno-Noguer[1] Juan Andrade-Cetto[1] François Fleuret[2,3]

[1]*Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain*
[2]*Computer Vision and Learning group, Idiap research institute, Martigny, Switzerland*
[3]*École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*
apenate@iri.upc.edu, fmoreno@iri.upc.edu, cetto@iri.upc.edu, francois.fleuret@idiap.ch

***Abstract*—We introduce LETHA (Learning on Easy data, Test on Hard), a new learning paradigm consisting of building strong priors from high quality training data, and combining them with discriminative machine learning to deal with low-quality test data. Our main contribution is an implementation of that concept for pose estimation. We first automatically build a 3D model of the object of interest from high-definition images, and devise from it a pose-indexed feature extraction scheme. We then train a single classifier to process these feature vectors. Given a low quality test image, we visit many hypothetical poses, extract features consistently and evaluate the response of the classifier. Since this process uses locations recorded during learning, it does not require matching points anymore. We use a boosting procedure to train this classifier common to all poses, which is able to deal with missing features, due in this context to self-occlusion. Our results demonstrate that the method combines the strengths of global image representations, discriminative even for very tiny images, and the robustness to occlusions of approaches based on local feature point descriptors.**

***Keywords*-pose estimation; low resolution; boosting;**

I. Introduction

The problem of 3D pose estimation is at the forefront in computer vision research. It consists of identifying the position and orientation of the camera, given a test image, with respect to the observed model. The problem has been addressed from either purely geometric or machine learning perspectives. Geometric methods initially use training data to build a 3D model, and then search for the 2D-to-3D correspondences that best align interest points in the test image with the 3D model [13]. Machine learning approaches on the other hand, annotate training imagery with discrete locations in the pose manifold, and then search globally for this pose-annotated matching of appearance, without resorting to full 3D reconstruction of the object [7], [16], [26]. The advantage of the global methods is that they are less sensitive to precise localization of individual features, which makes them more robust to image degradations than local geometric methods. But in contrast, global methods are not generally robust to occlusions. In addition, they often require splitting the pose space into several classes, and train specific classifiers for each of them, limiting the precision of the estimated pose to that of the granularity between classes and losing the correlation between neighboring poses.

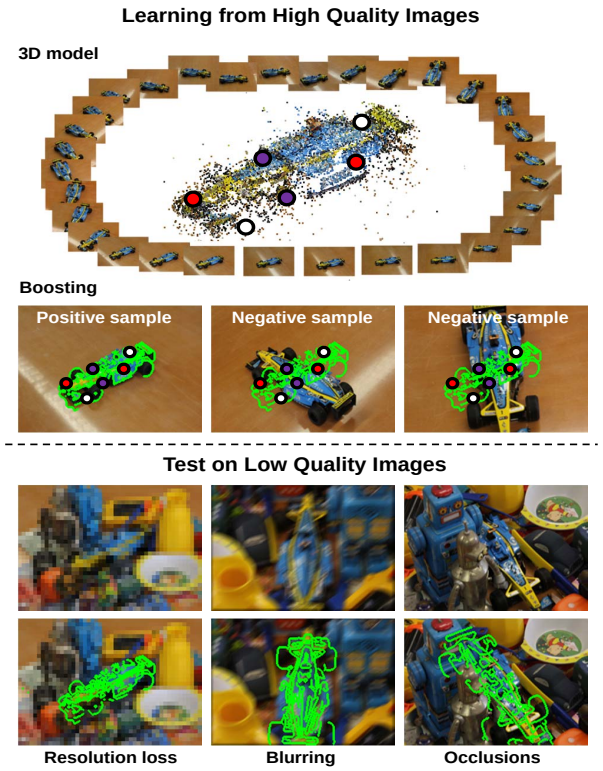

Figure 1. LETHA uses high-quality images to train a classifier that is tested on low quality data. The top frame shows the built 3D model, one positive and two negative training image-pose pairs. The fact that features are indexed with pose is indicated by the same location on the three training images of the projection of point pairs, and by the object contour projected in green. Observe that detecting the F1 car in the test images at the bottom frame is even difficult for the human eye due to different artifacts. The green contour indicates the pose estimated by our approach.

We propose the LETHA approach (Learning on Easy data, Test on Hard), that combines the strengths of both geometric and machine learning methods for estimating 3D object pose. We use high-definition training images to create

a 3D model of the object, and from it devise a pose-indexed feature extraction scheme that bind image quantities to the object pose. These features are then combined into strong priors for each training pose. Since we focus on one single object, the priors we build capture the variability of the appearance and generalize to test images with severe artifacts in a maner similar to that of [9], [4]. These approaches, though, again rely in the fact that similar local features appear in both training and test images. We get rid of this requirement by building specific priors for each training pose, in which we exactly know where the features should appear in the test image, hence no point of interest matching is needed. A *single* classifier, common to all the poses, is trained from these pose-indexed feature vectors. We use a procedure able to cope with incomplete feature vectors, a situation that occurs during self occlusion, allowing us to evaluate the classifier, even when some features have been wiped out or corrupted by image artifacts such as loss of resolution, motion blur or partial occlusions.

During test, given a low quality input image and a hypothetical pose to assess, the pose-indexed feature vector is computed similarly, without the need to detect and match points of interest, and fed to the classifier. The object's pose is estimated by measuring the classifier response for all the poses seen in the training images, and then refined by resampling around those poses with maximal classifier response. We insist on the fact that since this optimization is done by visiting multiple poses systematically, we do not need to match points of interest, or perform any type of fragile matching prior to using our predictor.

As shown in Fig. I, our method is able to estimate the pose even in the presence of severe image artifacts such as motion blur and occlusion. As we will demonstrate in the experimental section, LETHA compares favorably against geometric approaches based on SIFT and DBRIEF features, and also against global approaches based on Bag of Features descriptors [25], GIST [15] and PCA cross-correlation [23].

Table I
NOTATION

u	Image.
ξ	An hypothetical target pose to be tested
$\tilde{\mathbb{R}}$	Set of real numbers extended with the value $n.a.$
T	Nb. of high quality training images.
u_t^*	t-th training image.
R^+, R^-	Nb. of positive and negative samples for Boosting.
Q	Nb. of 3D points in the model.
p_q	q-th 3D point in the object model.
$\Pi(p, \xi)$	Projection of point p in the image for object's pose ξ.
$\Psi(u, \xi)$	Pose-indexed feature vector on image u for pose ξ.
D	Feature vector size.
Φ	Trained classifier.
M	Nb. of stumps in Φ.
σ	Threshold function extended to $n.a.$
ρ_m, ω_m	m-th stump threshold and weight.

II. Related Work

3D pose estimation methods may be roughly split in those techniques relying on local image features that purely use geometric relations to compute the pose; and methods that compute global descriptors of the image and resort to machine learning tools to estimate the pose.

Local approaches use feature point descriptors to estimate 2D-to-3D correspondences between one input image and one or several reference images registered to a 3D model of the object. PnP algorithms such as the EPnP [13], [18], [5] are then used to enforce geometric constraints and explicitly solve for the pose parameters. On top of that, robust RANSAC-based strategies [2], [14], [19] can be used both to speed up the matching process and to filter outlier correspondences. Yet, while these methods provide very accurate results, they require both the reference and input images to be of high quality, such that local features can be reliably and repetitively extracted. As we will show in the results section, these methods are not applicable for the level of image artifacts we consider in this paper.

By contrast, approaches relying on global descriptions of the object are less sensitive to a precise localization of individual features. These methods typically use a set of training images acquired from different viewpoints to statistically model the spatial relationship of the local features, either using one single detector for all poses [8], [10], [21] or a combination of various pose-specific detectors [16], [17], [22], [26]. Another alternative is to bind image features with poses during training and have them vote in the pose space [7]. These approaches, though, focus on recognizing instances of a generic class and are not designed to deal with image content different from that in the training set.

Among the methods that compute a global descriptor of the image, we find some holistic representations that do not require extracting points of interest. For instance, the GIST [15] descriptor encodes sustained overall orientation of straight edges on images, rather than localized features. This descriptor is conceived more as class descriptor than as a unique sample identifier, and is not generally robust for discriminating between poses, mainly because it is built using only 2D intensity data, disregarding visibility information of the 3D model. The same applies to the PCA cross-correlation, used in [23] as a similarity measure between tiny images. As it will be shown in the results section, considering visibility constraints in our pose-indexed feature vectors brings a remarkable advantage of our approach against such global descriptors, especially under occlusions. This is because, to account for occlusions, we devote a special treatment to missing data in our feature vector, giving a comprehensive solution to self-occlusions.

III. THE LETHA APPROACH

In this section we describe an implementation of the LETHA learning paradigm, applied to the estimation of the pose of an object in low-quality images.

First, as described in § III-B, we generate a 3D cloud model of the object, from which we derive a pose-indexed feature extraction scheme able to compensate for pose changes. Second, as described in § III-C, these features are combined into a *single* classifier common to all the poses. To handle weak learners abstaining because of self-occlusion, we use a boosting procedure, dubbed AbstainBoost. The search for the optimal pose is a coarse-to-fine process, as described in § III-D. We first visit exhaustively the poses met in the training set, and then visit more densely around the most promising hypotheses by generating synthetically perturbed poses in their neighborhoods.

A. Motivation and summary of the approach

Our overall approach consists of reformulating the estimation of the pose of the object of interest in a framework similar to the sliding-window approach for detection: we visit many "poses", and estimate for each a matching score with a *single* trained predictor. The key idea is that the extraction of the features alleviates the training of that single predictor by handling geometrical invariance.

1) Standard detection with a sliding window: For the sake of simplicity, consider first the sliding-window approach for face detection. Given an image u, it visits a large number R of sub-windows, each defined by a location in the image plane and a scale, and for each of these "hypothetical poses" $\{\xi_1, \ldots, \xi_R\}$, it extracts a vector of features $\Psi(u, \xi)$ in the corresponding sub-window, such as the responses of linear filters translated and scaled according to ξ, and feed them to a predictor Φ, such as an SVM or a Boosted linear predictor. The response $\Phi(\Psi(u, \xi))$ should be positive if a face is present there, negative otherwise.

The central idea is that the *same* predictor Φ is used for every window. The way the feature responses are computed ensures that Φ it does not have to cope with invariance to translation or scale.

A remarkable property of this approach, as noticed in [6], is that the "windows" do not really exist. What defines the overall process is (a) a set of poses $\{\xi_1, \ldots, \xi_R\}$, and (b) a procedure which to compute a "pose-indexed" feature vector $\Psi(u, \xi)$ for any image u and pose ξ, which accommodates the perturbations due to the pose. These two components are used to produce the training feature vectors to learn Φ, and the test feature vectors to use during detection.

2) Extension to general 6D poses: We can generalize the same approach to rigid objects, in which case the pose is 6D. For this, Ψ should extract quantities in the image at locations corresponding to 3D points fixed in the object reference frame, and projected in the image according to the hypothetical 6D pose to test.

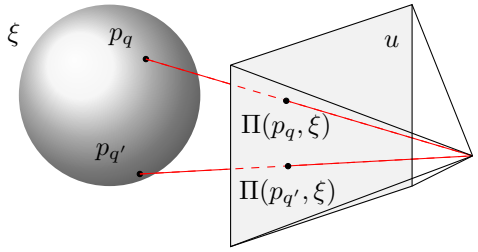

Figure 2. Each individual pose-indexed feature is computed as the difference between the gray levels at locations $\Pi(p_q, \xi)$ and $\Pi(p_{q'}, \xi)$, corresponding to the projections of the points p_q and $p_{q'}$ into the image plane u, for the hypothetical object pose ξ. If one of the points is not visible due to self-occlusion, the feature value is $n.a.$

Our algorithm goes one step further and *learns* from data both the set of poses $\{\xi_1, \ldots, \xi_R\}$, and the mapping Ψ. Given high-definition training images, we estimate the camera positions from which we build the set of poses $\{\xi_1, \ldots, \xi_R\}$, and a 3D model of the object from which we construct the functional Ψ. In practice, this Ψ computes quantities in the image at locations corresponding to points physically on the object. We extract features with this Ψ, and as in the standard sliding-window case, we train a single predictor Φ common to all the poses.

B. Learning pose-indexed features Ψ

The first step to learn Ψ from the high-definition training images $u_1^*, \ldots, u_T^*$ is to build a 3D cloud model of the object. We use Bundler [20], a SfM system that matches SIFT key-points through iterative bundle adjustment. It generate a dense family of Q points $p_q \in \mathbb{R}^3$ laying on the object's surface, and an estimate of each training image viewpoint pose, from which we derive the object pose in the observer's referential $\xi_t^* \in \mathbb{R}^3 \times \mathrm{SO}(3), \quad t = 1, \ldots, T$.

Then, for each gray-scale image u, and for any pose ξ, let $\Pi(p, \xi) \in \{1, \ldots, W\} \times \{1, \ldots, H\} \cup \{n.a.\}$ denote the projection into the image plane of u of the point p laying on the 3D model surface, with W and H being the image width and height, respectively. This projection will take the value $n.a.$ when the point is hidden due to self-occlusion. For any pair of point indexes $(q, q') \in \{1, \ldots, Q\}^2$, we define a pose-indexed feature as the difference between the image intensities at the two projected points (see Fig. 2):

$$\Psi_{q+Qq'}(u, \xi) = u(\Pi(p_q, \xi)) - u(\Pi(p_{q'}, \xi)) \ , \qquad (1)$$

which takes the value $n.a.$ if either one of the projections is $n.a.$ From these features, we define a full pose-indexed feature vector of dimension $D = Q^2$. This whole feature vector is never actually computed. During training, only a random subset of features is evaluated, and during test, the number of features actually evaluated is equal to the number $M << D$ of stumps in the predictor Φ.

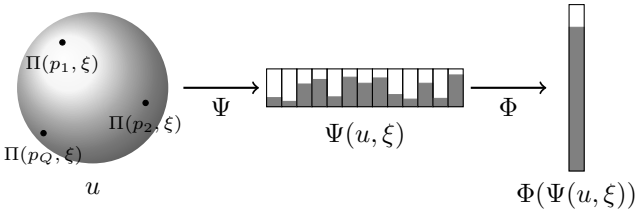

Figure 3. To evaluate the classifier on a test image u, for an hypothetical pose ξ, the algorithm first computes the feature vector $\Psi(u, \xi)$, and then the response of the predictor $\Phi(\Psi(u, \xi))$.

C. Training Φ with AbstainBoost

We want to train a predictor Φ to evaluate from the feature vector $\Psi(u, \xi)$, whether the image / pose pair (u, ξ) is consistent. That is, whether the object of interest is visible in u with the pose ξ.

1) Stumps and training set: We choose as predictor a linear combination of decision stumps:

$$\Phi(\psi) = \sum_{m=1}^{M} \omega_m \, \sigma(\psi_{d_m}, \rho_m) \ , \tag{2}$$

where ψ_{d_m} is the d_m-th feature of the pose-indexed feature vector ψ, ρ_m is the stump threshold, and ω_m is the stump weight, and all these parameters are chosen during training.

As stated in § III-B, features may take the value $n.a.$, to account for self-occlusion. We use a thresholding function that sends the $n.a.$ values to 0:

$$\sigma(z, \rho) = \begin{cases} 0 & \text{if} \quad z = n.a. \\ -1 & \text{if} \quad z < \rho \\ 1 & \text{if} \quad z \geq \rho \end{cases} \tag{3}$$

To build the training set, negative samples are generated by computing, for every single training image u_t^*, the pose-indexed feature vectors using the poses from other training images which have a relative difference with ξ_t^* greater than 10%. Positive samples are obtained by computing the pose-indexed feature vector for poses around ξ_t^* (i.e. we add a 1% relative noise to every component of the pose). The number R^+ of positive samples is a constant factor of the number R^- of negative samples (i.e., 30%).

Following the parallel with the sliding-window face detection, positive samples are taken "around" the actual location of every face, and negative samples are taken "far from" any face.

2) AbstainBoost: A classical method to build a linear combination of stumps from a training set is Adaboost, which selects stumps one after another to reduce the exponential loss in a greedy manner [11]. The standard derivation of this procedure relies on all the weak learners having the same L^2 norm. If we define $W_\tau = \sum_{n:y_n h(\psi_n)=\tau} \exp(-y_n \Phi(\psi_n))$, where $y_n \in \{-1, 1\}$ is the label of the n training samples, and ψ_n the corresponding feature vector, Adaboost chooses weak learners h maximizing $|W_{+1} - W_{-1}|$. However, as stated in Eq. (3), we have to deal with zero-valued responses, hence weak-learners of various norms. Relying on the inner product between the weak-learner's responses and the sample weights as an indication of "good direction" in the functional space, as Adaboost does, is incorrect and leads in practice to weak learners with fewer zero responses, even if they are often incorrect, because they have larger norm. With weak learners taking values in $\{-1, 0, 1\}$, it can be derived analytically that the optimal weak learner – i.e. the one inducing the maximum reduction of the exponential loss when added with its optimal weight ω – is the one maximizing $|\sqrt{W_{+1}} - \sqrt{W_{-1}}|$..

This is a natural derivation of Adaboost, the framework given here is analogous to Blum's "specialist" model of online learning [1]. For clarity we will refer to it as the AbstainBoost procedure. It can find the best stump's threshold in a time linear with the number of samples after they have been sorted according to the feature's value, and the optimal weight remains $\omega_m = \log \sqrt{W_{+1}/W_{-1}}$. This use of Adaboost is similar to the GrowRule operation in the Slipper algorithm [3], with the main difference that it allows to directly select signed abstaining decision stumps, instead of being applied to a greedy construction of Boolean disjunctions.

To summarize: given the training set, and the stumps defined on Eq. (3), the learning procedure consists of M AbstainBoost iterations, each one sampling at random several feature indexes $1 \leq d \leq D$, and keeping the one that maximizes the abovementioned score. The corresponding stump is then added to the strong classifier, and the process is re-iterated.

D. Coarse-to-fine pose estimation for testing

The test proceeds in a two-step "coarse-to-fine" manner, visiting first the poses seen during training, and then focusing on the best ones by visiting another set of poses generated in their neighborhoods. For each visited pose, the classifier response is computed as depicted in Figure 3.

More precisely the process first loops through the T training poses $\xi_1^*, \ldots, \xi_T^*$, and for each, it projects the Q object model points onto the image plane, and creates the pose-indexed feature vector $\Psi(u, \xi_t^*)$. This feature vector is used to evaluate the classifier response $\Phi(\Psi(u, \xi_t^*))$, and the G poses with the highest responses are retained

In a second step, the final pose is refined by reevaluating the classifier on a set of poses generated synthetically in the neighborhoods of the best G poses retained. We first set a hyper-box around each one of the G best poses by defining a minimum and maximum value for each single component, using the pose itself and its two closest neighbors, with an additional 10% relative margin on each component (see Fig. 4). We sample uniformly Z poses is in each box, and

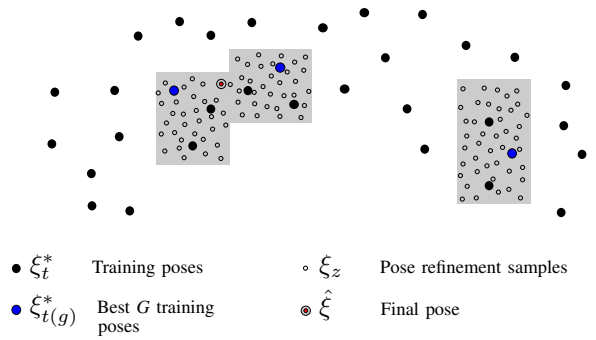

Figure 4. Refining the final pose. A bounding box in pose space is computed around each of the G poses that have maximal classifier response and their two nearest neighbors. Z new poses are uniformly sampled inside these boxes, keeping as final pose that with maximum classification score.

evaluate the classifier for each. The final pose $\hat{\xi}$ is the one with maximum response.

Note that the core property of our algorithm is that since we perform measurements at locations *recorded on the training images*, we do not require any point of interest detection in the test phase, since we know where the image intensities should be measured for each hypothetical pose we test. This makes the algorithm appealing for low-resolution or severely corrupted images.

IV. Experiments and Results

We next describe our experiments: the datasets we use for the evaluation, the competing approaches, the parameters used for learning the LETHA classifier, and the results.

A. Datasets

We use four datasets for evaluation (see Fig. 1 and 5 for some examples). In the first 2 datasets half the images were used for training and the other half for testing. Internal camera parameters for each image are known in all datasets.

- **Caltech dataset**: Objects from the *CalTech Turntable* dataset [12]. 360 images per object, in a controlled environment with constant lighting and textureless background.
- **Cars dataset**: Sequence used in [7]. 21 different cars observed under 68 viewpoints on average. Small number of instances per class, which makes learning difficult.
- **Sagrada Familia dataset**: Images with strong lighting changes and mild occlusions in two sets taken around the building, at different daytimes and with different weather conditions. 317 training images (taken on a sunny day) and 210 testing images (taken on a cloudy day). Occlusions due to pedestrians, buses and trees.
- **F1 dataset**: Training and test images were generated separately. It contains 317 calibrated training images, showing a F1 model car on an empty table, and 336 test images of the same F1 model car but in a heavy cluttered environment.

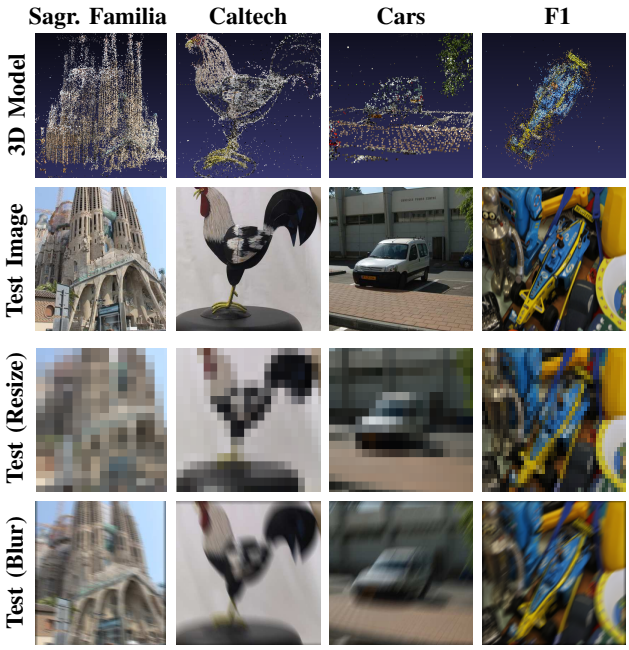

Figure 5. 3D model and sample images used in our tests. The images shown correspond to one original test image for each dataset along with the maximum point of degradation for resolution loss and motion blur reached in our experiments.

B. Baselines

Our first choice for a baseline was a purely geometric method relying on local feature point extraction with RANSAC-based geometrical pose estimation. Given a high definition 718×480 image of the Sagrada Familia dataset, and its closest pose-correspondent 143×96 image, we extracted points of interest in both images using DBRIEF [24], computed the number of inliers after matching features and estimated the pose transform using EPnP[13] with RANSAC. Due to the large amount of mismatches and localization biases of the same 3D in both images, even after 50,000 RANSAC iterations (which took 5 minutes to compute in a standard PC) the algorithm was not able to converge to a correct pose. Obviously the problem would be magnified if, for a given test image, all training images had to be evaluated, and thus we discarded RANSAC-based approaches from subsequent analysis.

The baselines we chose are state of the art methods of both geometrical and global approaches:

- **PCA-NCC**: PCA normalized cross-correlation [23] is a good candidate, as it does not require an outlier rejection stage to compare a pair of images.
- **GIST**: Gist descriptor [15] uses information of the entire image, and as suggested by [23] it is an appropriate approach to compare very tiny images.
- **BoF**: As a representative of the methods that build a global descriptor of the image from local features we used a Bag

of SIFT Features (BoF) [25].

• **DBRIEF**: We used the average confidence of individual DBRIEF matches [24]. It uses a similar scheme as ours, it trains a dictionary of features by learning the appearance changes over a 3D model. This is done to ensure robustness to 3D deformations with the same intention as LETHA.

C. Training and testing with LETHA

Using Bundler, we built the 3D models for each dataset. The size of these models ranges from about 1×10^5 points for the Sagrada Familia dataset down to 2×10^4 points in the F1 dataset. For the Cars dataset we built a different 3D model for each object class.

Training images are initially convolved by Gaussian filters with standard deviation of 2 pixels. Then, following the methodology described in § III-C1, for each of the T training poses, we generate R^+ positive and R^- negative samples. Training images are initially convolved by Gaussian filters with standard deviation of 2 pixels. Then, following the methodology described in § III-C1, for each of the T training poses, we generate R^+ positive and R^- negative samples, according to the procedure described in § III-C1. This results, for each training image, in a total of 52 samples for the Cars dataset, 240 for the Sagrada Familia dataset, 300 for the Caltech dataset, and 412 for the F1 dataset. The full sample set used to train the predictor Φ has a size ranging from $10,000$ samples to $350,000$ samples, out of which $1/4$-th are positive samples and $3/4$-ths negative ones.

D. Results

In all experiments, we compute the pose of a corrupted test image with each of the algorithms. For all competing methods, the estimated pose will be the one of the most similar training image. For LETHA it is computed as described in § III-D. Let $\hat{\xi} = (\hat{\mathbf{q}}, \hat{\mathbf{t}})$ be that estimated pose, where $\hat{\mathbf{q}}$ is a normalized quaternion representing the rotation and $\hat{\mathbf{t}}$ the translation vector. Similarly, let $\xi_{\text{true}} = (\mathbf{q}_{\text{true}}, \mathbf{t}_{\text{true}})$ be the ground truth pose of the test image. As in [13], relative rotation and translation errors is computed as $\mathbf{E}_{\text{rot}} = \|\mathbf{q}_{\text{true}} - \hat{\mathbf{q}}\|/\|\hat{\mathbf{q}}\|$ and $\mathbf{E}_{\text{trans}} = \|\mathbf{t}_{\text{true}} - \hat{\mathbf{t}}\|/\|\hat{\mathbf{t}}\|$, respectively. In all discussed experiments we compute the median error of all testing images over different configurations.

Note that LETHA refines the pose by sampling around the training poses with highest scores. Yet, given a training pose, the reduction in pose error when using this fine estimation is very small. The real benefit of re-sampling around a few training poses, is that the final chosen pose may be the result of sampling around a training pose which initially did not have the highest score. We show examples of pose retrieval on test images in Fig. 7.

1) Resolution loss and blurring: We evaluated all methods in two different situations: reduction of the image size and motion blur (see Fig. 5). Let us first focus on the Caltech and Cars datasets for which the amount of illumination changes or occlusions produced by external objects does not exist or is relatively small.

The first two rows of Fig. 6 summarize these results. For the "size reduction" experiment both PCA-NCC implementations and our approach show high robustness[1]. For instance, in the Sagrada Familia dataset this means that the algorithms are capable of finding the right pose for a test image as small as 14×10 pixels compared to the 718×480 size of a training image. The performance of our approach and of PCA-NCC degrade for larger reduction sizes. Note that PCA-NCC, despite being a relatively simple approach, takes advantage of the fact that it uses all pixels in the image. The rest of methods that either rely on combination of local features (BoF or DBRIEF), or orientations of straight edges (GIST), generally rapidly fail for moderate reductions in size, when these features are prone to disappear.

The performance in the "motion blur" experiment is similar. Both PCA-NCC and LETHA clearly outperform other techniques in the Sagrada Familia and Cars dataset. In the Caltech dataset, though, LETHA is consistently more robust than PCA-NCC.

2) Illumination changes: Our second series of experiments uses the Sagrada Familia dataset. Test images contain strong illumination changes from the training dataset since they were captured at different days, and with different weather conditions. This strong changes in appearance make PCA-NCC and BoF unable to handle even the non-corrupted images. GIST is able to handle the non-corrupted images but breaks when small changes are introduced. We also see that DBRIEF is able to handle quite well the object as it is highly textured. The difference between BoF and DBRIEF, both based on feature points, is due to BoF taking into account all the features in its histograms, which introduces a significant amount of noise. DBRIEF on the other hand takes into account mainly just the best matches, which in a higly textured object like the Sagrada Familia, is prone to have at least a couple of good matches even when changes in illumination are present. We are able to cope with illumination changes because we use features based on the difference of intensity between points and not on the absolute intensity values.

3) Occlusions: Let us now focus on the experiments for the F1 dataset depicted in the last row of Fig. 6. As shown in Fig. 1, the test images contain strong occlusions of the object which were not included in the training set. On top of this, we also considered the image degradation artifacts used above. In this situation neither the methods that use information of the entire image (PCA-NCC and GIST), nor the methods based on local features (BoF and DBRIEF), are able to succeed. LETHA, in contrast, exploits all its properties to yield robust results. On the one hand, the fact

[1]Relative pose errors below 10% are comparable to those that would be obtained using a purely geometric method, such as the EPnP [13] when using high resolution images with no degradation.

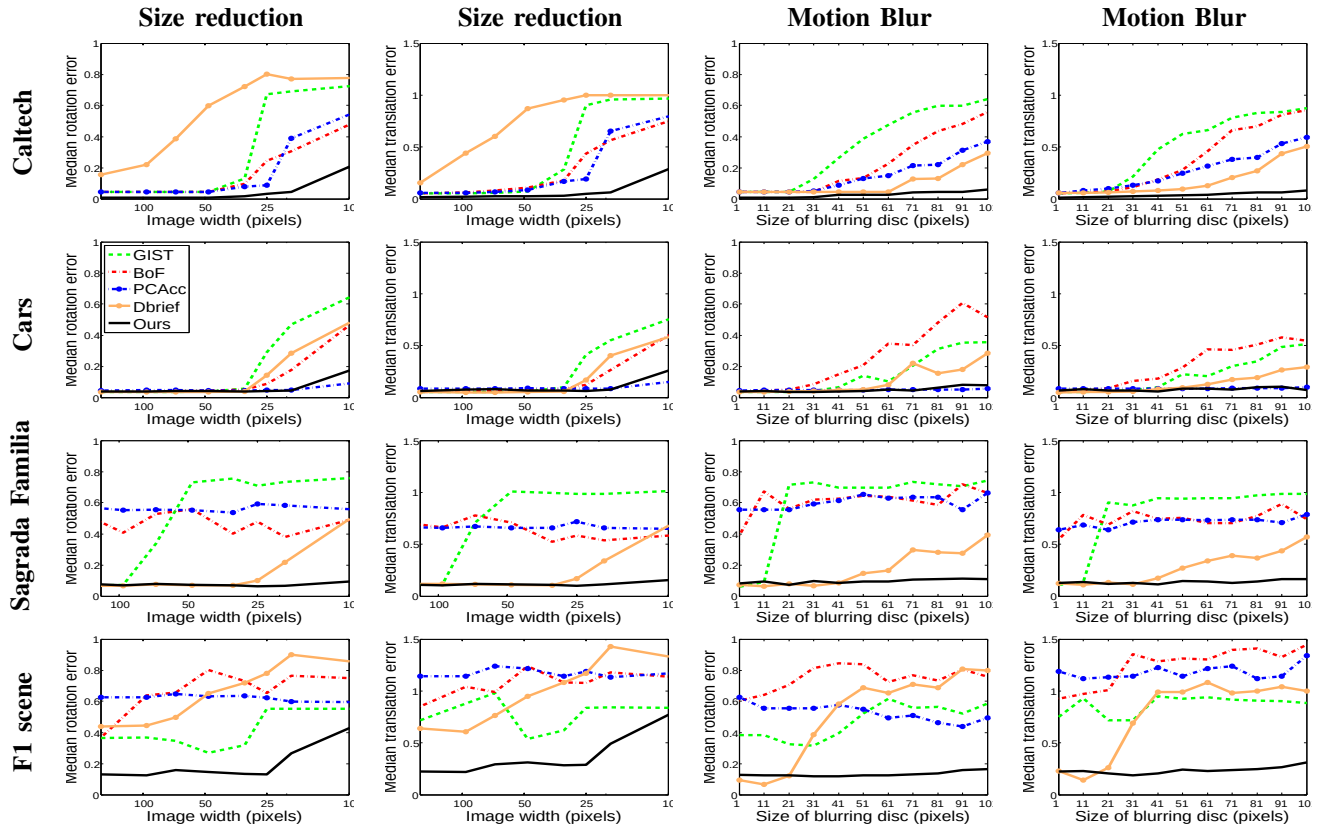

Figure 6. Pose estimation error of LETHA and other approaches in experiments with severe degradations of image size and motion blur. In addition the last row depicts the results on the F1 dataset which contains strong occlusions.

that the location of the features for each pose is known in advance alleviates the problem of feature detection when the image is severely deteriorated. On the other hand, it also exploits the fact that for each view only local features that are not affected by self-occlusion are considered. This maximizes the chance of obtaining a large number of visible features, even when the target object is partially occluded by external objects.

4) Computational cost: Given a test image, the time to estimate the pose for the different experiments is shown in Table II. Note that all methods are about the same order of magnitude. DBRIEF is slow because we have used its MATLAB implementation. In addition, while it is fast in extracting the features, the process of matching a large number of them is slow.

	GIST	BoF	PCA-NCC	DBRIEF	LETHA
Sagr. Fam	1.19	0.12	0.27	8.30	1.64
Caltech	0.89	0.10	0.34	7.90	1.56
Cars	0.67	0.19	0.38	2.09	0.82
F1	1.56	0.22	0.47	17.1	4.34

Table II
TIME (SECONDS) REQUIRED TO COMPUTE THE POSE OF AN INPUT IMAGE FOR ALL EXPERIMENTS AND METHODS. NOTE THAT BOF IS IMPLEMENTED IN C WHILE THE OTHER METHODS ARE IN MATLAB

V. CONCLUSION

We have proposed a new machine learning paradigm: Learning with high-quality data to be able to test with low quality data. The rationale behind this idea is that inference is possible only from clean data, or using a strong model, and that the latter can be inferred from the former.

From this general principle, and extending the concept of pose-indexed features to be able to learn them, we have derived a novel and very efficient algorithm for the specific problem of 3D pose estimation. As demonstrated on the test datasets, with sufficiently good training data, we obtain an extremely good estimate of the object pose, in very low resolution images, with illumination changes and with high levels of noise and occlusion.

This procedure is promising as a near-perfect solution to be used in controlled environments such as a factory. Our future work will aim at extending it to multi-target detection, richer poses, and class-level detection.

Figure 7. Pose estimation results represented as reprojected wireframes of the best matching candidate in the training data. Allthough our pose is better than the closest image from the training set, we do not directly project the 3D model point cloud because it would clutter the image in an unintuitive way.

ACKNOWLEDGMENT

This work has been partially funded by Spanish Ministry of Economy and Competitiveness under project PAU+ DPI2011-27510 and ERA-Net Chistera project ViSen PCIN-2013-047; and by the EU project ARCAS FP7-ICT-2011-28761; A. Penate-Sanchez is the recipient of a JAE-Predoc scholarship funded by the European Social Fund.

REFERENCES

[1] A. Blum. Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain. *Mach. Learn.*, 26(1):5–23, 1997.

[2] O. Chum and J. Matas. Matching with PROSAC - progressive sample consensus. In *CVPR*, 2005.

[3] W. W. Cohen and Y. Singer. A simple, fast, and effective rule learner. In *AAAI*, 1999.

[4] G. Fanelli, J. Gall, and L. van Gool. Real time head pose estimation with random regression forests. In *CVPR*, 2011.

[5] L. Ferraz, X. Binefa, and F. Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. In *Conference in Computer Vision and Pattern Recognition (CVPR)*, 2014.

[6] F. Fleuret and D. Geman. Stationary features and cat detection. *J Machine Learning Res.*, 9:2549–2578, 2008.

[7] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011.

[8] W. Hu and S.-C. Zhu. Learning a probabilistic model mixing 3D and 2D primitives for view invariant object recognition. In *CVPR*, 2010.

[9] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE T. Pattern Anal. Machine Intell.*, 28(9):1465–1479, 2006.

[10] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, 2010.

[11] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *NIPS*, 2000.

[12] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. *Int. J. Computer Vision*, 73(3):263–284, 2006.

[13] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative $O(n)$ solution to the PnP problem. In *ICCV*, 2007.

[14] F. Moreno-Noguer, V. Lepetit, and P. Fua. Pose priors for simultaneously solving alignment and correspondence. In *ECCV*, 2008.

[15] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. Computer Vision*, 42(3):145–175, 2001.

[16] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[17] N. Payet and S. Todorovic. From contours to 3D object detection and pose estimation. In *ICCV*, 2011.

[18] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2387–2400, 2013.

[19] A. Penate Sanchez, E. Serradell, F. Moreno Noguer, and J. Andrade Cetto. Simultaneous pose, focal length and 2d-to-3d correspondences from noisy observations. In *Proceedings of the British Machine Vision Conference*, 2013.

[20] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring image collections in 3D. *ACM T. Graphics*, 25:835–846, 2006.

[21] 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 *ICCV*, 2009.

[22] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiel, and L. V. Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[23] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE T. Pattern Anal. Machine Intell.*, 30(11):1958–1970, 2008.

[24] T. Trzcinski and V. Lepetit. Efficient Discriminative Projections for Compact Binary Descriptors. In *ECCV*, 2012.

[25] A. Vedaldi and B. Fulkerson. Vlfeat – an open and portable library of computer vision algorithms. In *ACM MM*, 2010.

[26] M. Villamizar, H. Grabner, F. Moreno-Noguer, J. Andrade-Cetto, L. V. Gool, and A. Sanfeliu. Efficient 3D object detection using multiple pose-specific classifiers. In *BMVC*, 2011.