Implicit Elastic Matching with Random Projections for Pose-Variant Face Recognition

John Wright*
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
jnwright@uiuc.edu

Gang Hua
Microsoft Live Labs Research
ganghua@microsoft.com

Abstract

We present a new approach to robust pose-variant face recognition, which exhibits excellent generalization ability even across completely different datasets due to its weak dependence on data. Most face recognition algorithms assume that the face images are very well-aligned. This assumption is often violated in real-life face recognition tasks, in which face detection and rectification have to be performed automatically prior to recognition. Although great improvements have been made in face alignment recently, significant pose variations may still occur in the aligned faces. We propose a multiscale local descriptor-based face representation to mitigate this issue. First, discriminative local image descriptors are extracted from a dense set of multiscale image patches. The descriptors are expanded by their spatial locations. Each expanded descriptor is quantized by a set of random projection trees. The final face representation is a histogram of the quantized descriptors. The location expansion constrains the quantization regions to be localized not just in feature space but also in image space, allowing us to achieve an implicit elastic matching for face images. Our experiments on challenging face recognition benchmarks demonstrate the advantages of the proposed approach for handling large pose variations, as well as its superb generalization ability.

1. Introduction

Human face recognition remains one of the most active areas in computer vision, due to its many applications, both in traditional security and surveillance scenarios as well as in emerging online scenarios such as image tagging and image search. While considerable algorithmic progress has been made on well-aligned face images, pose variation remains an obstacle to deployable robust face recognition in real-life photos. Figure 1 illustrates the difficulty: while popular face detectors such as the Viola-Jones algorithm [26] produce rough localizations of the face, significant misalignment remains even after aligning the eye locations using an automatic eye detection algorithm. When applied in this setting, classical algorithms [25, 2] designed for well-aligned face images break down.

Figure 1. **Misalignment in real face images.** Faces detected by the Viola-Jones face detector and aligned using a neural-network based eye detector. Even after these rectification steps, significant local discrepancies due to pose variations still remain.

The ability of different approaches to cope with face pose and misalignment can be roughly determined by the amount of explicit geometric information they use in the face representations. At one end of the spectrum are methods based on full three-dimensional face representations [3]. Such representations allow recognition across the widest possible range of poses, at the cost of system and computational complexity. Deformable two-dimensional models such as active appearance models [5, 9] offer an intermediate representation, as a deformable mesh plus texture. The elastic bunch graph matching (EBGM) approach of [29] utilizes a similar representation of face geometry and deformation, but restricts the texture representation to a small set of high-dimensional feature vectors, such as Gabor jets, located at the vertices of the mesh. In testing, the mesh is deformed so that these features best match the input face image, subject to a penalty on deformation complexity.

Speed improvements over EBGM can be realized by dropping the geometric constraint and instead matching

*This work was performed while John Wright was an intern at Microsoft Live Labs Research. The authors thank Dr. Michael Revow for building the eye detector used in this work.

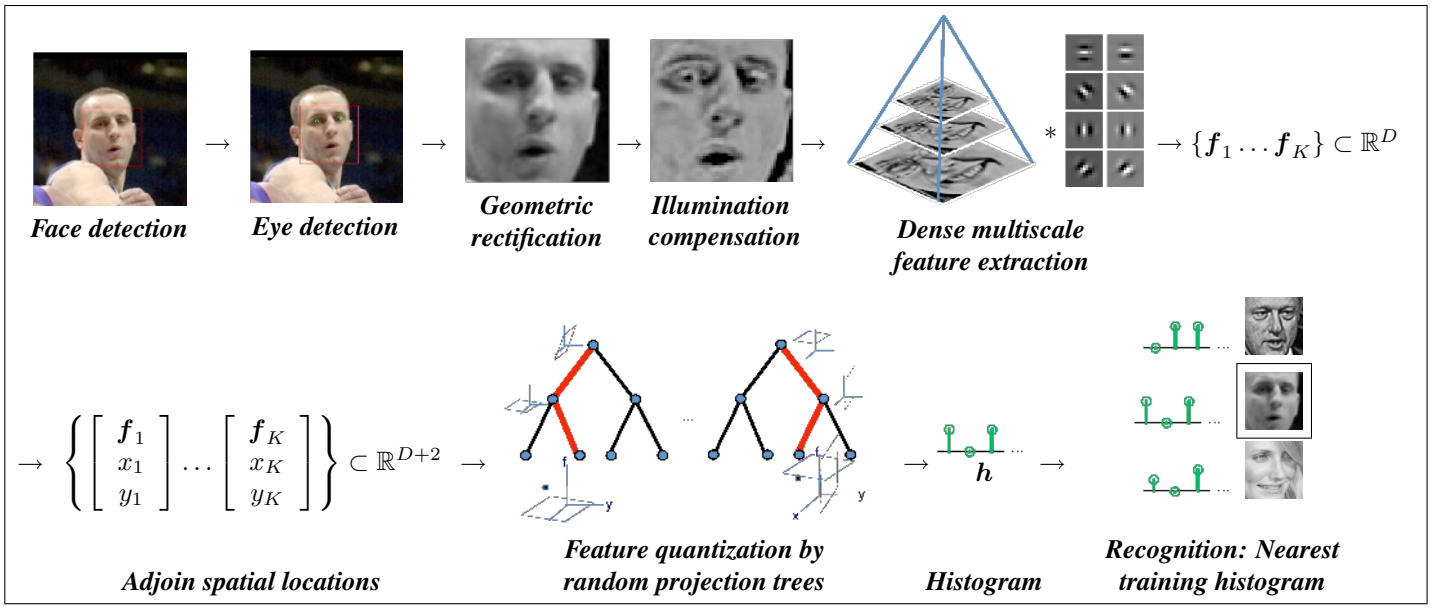

Figure 2. **Our pipeline.**

approximately-invariant feature descriptors such as SIFT keys [16] between the test image and each image in the database [17]. The smoothness of the face makes it a somewhat unnatural candidate for feature matching, however, since it limits the number of repeatable feature points that can be reliably extracted. Finally, fully 2D methods such as Laplacian Eigenmaps [11] can be applied to learn linear projections that respect any manifold structure present in the training data. While these algorithms are extremely fast in testing, characterizing the nonlinear structure of face images under pose and misalignment is difficult when only a few training samples are available. Moreover, the performance of such discriminative linear embedding methods [2, 11] is highly dependent on the specific dataset used for training: the learned feature transformation does not generalize to new faces or new datasets.

As demonstrated in Figure 1, even the best 2D or 2.5D alignment algorithms are intrinsically imperfect, due to pose, self-occlusion, ect. The difficulty of coping with such variations directly from 2D data is one of the factors behind the popularity of high-dimensional near-invariant features in image classification [16, 19, 28]. Unlike the explicit deformable matching performed by EBGM, these methods perform an *implicit* feature matching by quantizing the features and comparing statistics of the quantizations (e.g., histograms). A number of quantizer architectures have been investigated, including K-means trees [19] and randomized K-D tree variants [22, 15]. More recently, efforts have been made to couple the learning of the quantization scheme and the subsequent classifier [31].

However, intuition from high-dimensional geometry [14, 1] suggests that as long as the feature dimension is large enough, randomized quantization schemes with only very weak data dependence may already be sufficient to achieve good performance. For example, Dasgupta and Freund [6] prove that for data with low-dimensional structure embedded in a high-dimensional ambient space, inducing a tree by splitting along randomly chosen directions yields an efficient quantizer: the expected cell diameter is controlled by the intrinsic dimension of the data, irrespective of the ambient dimension.[1] This property is especially appealing for the high-dimensional feature vectors common in computer vision, which often exhibit intrinsically sparse or low-dimensional structure.

In light of the above developments, this paper proposes a very simple, efficient algorithm for recognizing misaligned and pose-varying faces. Like bunch graph matching, the algorithm works with a set of high-dimensional image features, although our image features are more discriminative and invariant for matching [28]. In contrast to bunch graph matching, rather than searching for a globally optimal matching, the algorithm instead performs a "soft" or "implicit" matching by jointly quantizing feature values and the spatial locations from which they were extracted. The quantizer consists of a forest of randomized decision trees, in which each node acts on a random projection of the data. Because the trees are only weakly data-dependent, they exhibit good generalization in practice, even across very different datasets. This nice property is in contrast to many previous methods which perform strong supervised learning, such as SVM [30] or LDA [2], to obtain a distance

[1] For a d-dimensional submanifold of $\mathbb{R}^D$, the cell diameter at level L drops as $e^{-O(L/d)}$, rather than $e^{-O(L/D)}$.

metric from the training data, which do not generalize well to new face datasets.

In the rest of the paper, we begin with an overview of our face recognition pipeline in Section 2. Core components in our pipeline, such as local feature representation, joint feature and spatial location quantization using random projection trees, as well as our face recognition distance metric are discussed in details in Section 3, Section 4, and Section 5, respectively. In Section 6 we perform a number of simulations to investigate the effects of various parameters, and then perform large-scale experimental comparisons to a number of recent algorithms, across publicly available face datasets. Section 7 summarizes other possible extensions and some of our key observations of the proposed work. Finally, Section 8 concludes.

2. Face Recognition Pipeline

Figure 2 gives an overview of our system as a whole. The system takes as its input an image containing a human face, and begins by applying a standard face detector (such as Viola-Jones [26]). Eye detection is performed based on the approximate bounding box provided by the face detector. Our eye detector is a neural network based regressor whose input is the detected face patches. Geometric rectification is then performed by mapping the detected eyes to a pair of canonical locations using a similarity transformation. Finally, we perform a photometric rectification step that uses the self-quotient image [27] to eliminate smooth variations due to illumination.

In our pipeline, the resulting face image after geometric and photometric rectification has size 32 pixels $\times$ 32 pixels. From this small image, we extract an overcomplete set of high-dimensional near-invariant features, computed at dense locations in image space and scale. These features are augmented with their locations in the image plan and are then fed into a quantizer based on a set of randomized decision trees. The final representation of the face is just a sparse histogram of the quantized features. An IDF-weighted ℓ^1 norm is adopted as the final distance metric for the task of recognition. The entire pipeline is implemented in C++, and requires less than a second per test image on a standard PC. The following sections give more extensive implementation details for the critical steps: feature extraction, learning the quantizer for building representation for faces, and recognition.

3. Local Feature Representation

We extract a dense set of features at regular intervals in space and scale. Dense features allow us to guarantee that most features in the test image will have an (approximate) match in the corresponding gallery image, without having to rely on keypoint detection. In practice, we find it sufficient to form a Gaussian pyramid of images (properly rectified and illumination-normalized as described above) of size 32×32, 31×31, and 30×30. Within each of these images, we compute feature descriptors at intervals of two pixels. The descriptors are computed from 8×8 patches, upsampled to 64×64. The set of feature patches for a given input face image is visualized in Figure 3.

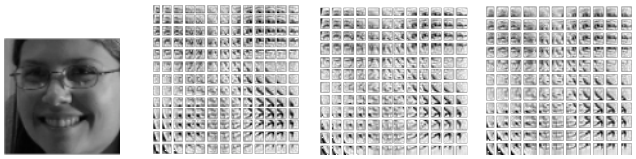

Figure 3. **Dense, multiscale patches.**

We compute a feature descriptor $\boldsymbol{f} \in \mathbb{R}^D$ for each patch. For most of the experiments in this paper, we use a $D = 400$-dimensional feature descriptor proposed in [28], and shown there to outperform a number of competitors on matching tasks. This descriptor, denoted T3h-S4-25 in [28], aggregates responses to quadrature pairs of steerable fourth-order Gaussian derivatives. The responses to the two quadrature filters are binned by parity and sign (i.e., even-positive, ect.), giving four responses (two of which are nonzero) at each pixel.[2] Four steering directions are used, for a total of 16 dimensions at each pixel. These 16-dimensional responses are aggregated spatially, in a Gaussian-weighted log-polar arrangement of 25 bins for an overall feature vector dimension of 400.

To incorporate loose spatial information into the subsequent feature quantization process, we concatenate the pixel coordinates of the center of the patch onto its feature descriptor, for a combined feature dimension of 402. Notice that we do not include scale information; we wish to be as invariant as possible to local scalings and it is perhaps inappropriate to treat such a coarse quantization of scale as a continuous quantity in the feature vector.

The total number of feature vectors extracted from each image is 457. Notice that this is a highly overcomplete representation of the fairly small (32×32) detection output. This expansion is conceptually similar to kernel tricks in machine learning, in which lifting low dimensional data into a high dimensional space allows very simple decision architectures such as linear separators (or here, even random linear separators) to perform very accurately.

In our current implementation, the vast majority of the computation is spent on this feature extraction step. This computational effort could be dramatically reduced by exploiting overlap between spatially adjacent feature loca-

[2] This thresholding tends to lead to sparse vectors $\boldsymbol{f}$, in which many bins are identically zero. Random projections are an especially appropriate tool for quantizing such vectors, since they are incoherent with the standard basis. In fact, one of the simplest theoretical examples in which random projections outperform standard k-d trees occurs when the data consist only of the standard basis vectors and their negatives [6].

tions, using ideas similar to [24].

4. Joint Feature and Spatial Quantization

The training phase of our algorithm begins with the set of all (augmented) features extracted from a set of training face images. We induce a forest of randomized trees, $T_1 \dots T_k$. Each tree is generated independently, and each has a fixed maximum depth h. At each node v of the tree, we generate a random vector $\boldsymbol{w}_v \in \mathbb{R}^{D+2}$ and a threshold

$$\tau_v = \text{median}\{\langle \boldsymbol{w}_v, \tilde{\boldsymbol{f}} \rangle \mid \tilde{\boldsymbol{f}} \in \mathcal{X}\},$$

corresponding to the binary decision rule

$$\langle \boldsymbol{w}_v, \cdot \rangle \gtrless \tau_v. \tag{1}$$

The training procedure then recurses on the left- and right-subsets $\mathcal{X}_L \doteq \{\tilde{\boldsymbol{f}} \mid \langle \boldsymbol{w}_v, \tilde{\boldsymbol{f}} \rangle \leq \tau_v\}$ and $\mathcal{X}_R \doteq \{\tilde{\boldsymbol{f}} \mid \langle \boldsymbol{w}_v, \tilde{\boldsymbol{f}} \rangle > \tau_v\}$. The random projection $\boldsymbol{w}_v$ is sampled from an anisotropic Gaussian

$$\boldsymbol{w}_v \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \sigma_{\boldsymbol{f}}^{-2} I_{D\times D} & \\ & \sigma_x^{-2} I_{2\times 2} \end{bmatrix}\right), \tag{2}$$

where $\sigma_{\boldsymbol{f}}^2 = \text{trace}\,\hat{\Sigma}(\boldsymbol{f})$ and $\sigma_x = \text{trace}\,\hat{\Sigma}(x, y)$, and $\hat{\Sigma}$ denotes the empirical covariance across the entire dataset. Notice that this choice of distribution is equivalent to reweighting the vectors $\tilde{\boldsymbol{f}}$ so that each segment (feature and location) has unit squared-ℓ^2-norm on average, and balances the fact that the feature vector is much higher-dimensional than the appended coordinates.

While the theoretical properties of randomized trees are appealing, in practice the performance can often be improved by sampling a number of random projections, and then choosing the one that optimizes a task-specific objective function, e.g., the average cell diameter [7]. Moreover, it is neither necessary nor feasible to save a unique $D+2$-dimensional vector $\boldsymbol{w}_v$ at each node v. Instead, we choose a dictionary of $W = \{\boldsymbol{w}^{(1)} \dots \boldsymbol{w}^{(k)}\}$ ahead of time, and at each node v set $\boldsymbol{w}_v$ to be a random element of W. This allows us to store only the index of $\boldsymbol{w}_v$ in W, and does not break the sample-path guarantees of [6]. For extremely large face databases, further computational gains can be realized via an inverted file structure, in which each leaf of the forest contains the indices of a list of training images for which the corresponding histogram bin is nonempty.

While it may seem like a minor implementation detail, the expansion of the features by x, y is actually critical in ensuring that the quantization remains as discriminative as possible while also maintaining robustness to local deformations. Because the quantizer acts in the joint space $(\boldsymbol{f}, x, y)$ it captures both deformations in feature value and domain, generating a set of linear decision boundaries in this space. Figure 4 (left) visualizes these quantization regions in the following manner: a feature descriptor $\boldsymbol{f}$ is extracted near the corner of the eye, at point x, y. This descriptor is translated to every point x', y' on the image plane. The intensity of the blue shading on the image (duplicated at bottom right) indicates the number of trees in the forest for which $(\boldsymbol{f}, x, y)$ and $(\boldsymbol{f}', x', y')$ are implicitly matched. Notice that the strongest implicit matches are all near the corner of the eye space, and also correspond in (feature) value to patches sampled near eye corners.

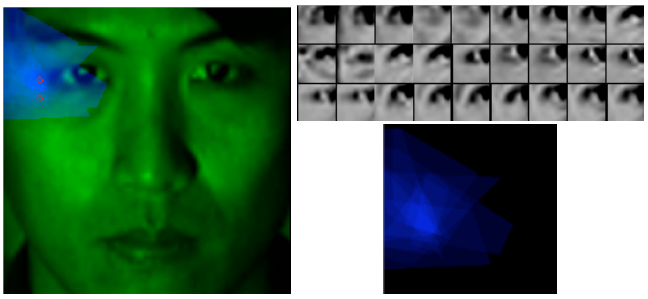

Figure 4. **Joint feature-spatial quantization.** Left: one bin 10 tree forest learned from the CMU PIE dataset. A feature $\boldsymbol{f}$ is extracted from the subject's left eye corner (x, y), and translated to various locations (x', y'). At each location, the blue intensity indicates the number of trees for which $(\boldsymbol{f}, x, y)$ and $(\boldsymbol{f}, x', y')$ are implicitly matched. Right: at top, a subset of patches that quantize to the same bin in least 3 trees. At bottom, number of bins. Notice that the quantizer restricts itself (softly) to the area around the left eye corner, and that most of the patches are eye corners.

This example also highlights the importance of having a forest rather than just a single tree: aggregating multiple trees creates a smoothing of the region boundary that better fits the structure of the data. We will further examine the effect of quantizer architecture in Section 6.

Algorithm 1: Tree induction (`rptree`)

Input: Augmented features $\mathcal{X} = \{\tilde{\boldsymbol{f}}_1 \dots \tilde{\boldsymbol{f}}_m\}$, $\tilde{\boldsymbol{f}}_i = (\boldsymbol{f}_i, x_i, y_i) \in \mathbb{R}^{D+2}$.
Compute feature and coordinate variances σ_f^2 and σ_x^2.
Generate $p \geq D+2$ random projections

$$W \sim_{iid} \mathcal{N}(\boldsymbol{0}, \text{diag}(\sigma_f^{-2} \dots \sigma_f^{-2}, \sigma_x^{-2}, \sigma_x^{-2})).$$

repeat k times
  Sample $i \sim \text{uni}(\{1 \dots p\})$.
  $\tau_i \leftarrow \text{median}\{\langle \boldsymbol{w}_i, \tilde{\boldsymbol{f}} \rangle \mid \boldsymbol{f} \in \mathcal{X}\}$
  $\mathcal{X}_L \leftarrow \{\tilde{f} \mid \langle \boldsymbol{w}_i, \tilde{f} \rangle < \tau_i\}$, $\mathcal{X}_R \leftarrow \mathcal{X} \setminus \mathcal{X}_L$.
  $r_i \leftarrow |\mathcal{X}_L|\,\text{diameter}^2(\mathcal{X}_L) + |\mathcal{X}_R|\,\text{diameter}^2(\mathcal{X}_R)$
end
Select the $(\boldsymbol{w}^*, \tau^*)$ with minimal r.
$\text{root}(T) \leftarrow (\boldsymbol{w}^*, \tau^*)$.
$\mathcal{X}_L \leftarrow \{\tilde{f} \mid \langle \boldsymbol{w}^*, \tilde{f} \rangle < \tau^*\}$, $\mathcal{X}_R \leftarrow \mathcal{X} \setminus \mathcal{X}_L$.
$\text{leftchild}(T) \leftarrow \texttt{rptree}(\mathcal{X}_L)$
$\text{rightchild}(T) \leftarrow \texttt{rptree}(\mathcal{X}_R)$
Output: T.

5. Recognition Distance Metric

The recognition stage of our algorithm is extremely simple. Each gallery and probe face image is represented by a histogram $\boldsymbol{h}$ whose entries correspond to leaves in $T_1 \dots T_k$. The entry of $\boldsymbol{h}$ corresponding to a leaf L in T_i simply counts the number of features $\boldsymbol{f}$ of the image for which $T_i(\boldsymbol{f}) = L$. Notice that each feature $\boldsymbol{f}$ contributes to k bins of $\boldsymbol{h}$; similar concatenation is used in [18].

There are many possible norms or distance measures for comparing histograms. We find consistently good performance using a weighted ℓ^1-norm with weightings corresponding to the inverse document frequencies (the so-called TF-IDF scheme [19]). More formally let $\mathcal{X} = \{\mathcal{X}_i\}$ be the set of all the training faces, and $\boldsymbol{h}_i$ be the quantization histogram of X_i, we have

$$d(\boldsymbol{h}_1, \boldsymbol{h}_2) \doteq \sum_j w_j |\boldsymbol{h}_1(j) - \boldsymbol{h}_2(j)|$$

$$w_j \doteq \log \frac{|\mathcal{X}|}{|\{\mathcal{X}_m : \boldsymbol{h}_m(j) \neq 0\}|} \tag{3}$$

where $|\cdot|$ denotes the cardinality of the corresponding set. The intuition of this IDF weighting is that quantization bins whose values appear in many face images should be down-weighted because they are less discriminative. Section 6 further investigates the appropriateness of this distance measure. Notice that this matching scheme has the ability to scale to large face dataset using similar inverted file architecture as in [19].

6. Simulations and experiments

In this section, we first investigate the effect of various free parameters on the performance of the system. We then fix the parameters and perform large-scale evaluations across several publicly available datasets.[3]

6.1. Effect of tree structure

Before performing large-scale recognition experiments, we first investigate the effect of various parameter choices on the algorithm performance. For these experiments, we use a subset of the CMU PIE [23] database, containing a total of 11,554 images of 68 subjects under varying pose (views `C05`, `C07`, `C09`, `C27`, `C29`).[4] A random subset of 30 images of each subject's images are used for training (inducing the forest) and the remainder for testing.

[3] Fixing the parameters helps avoid overfitting; however, further improvements in performance may be possible by tuning the algorithm for larger datasets.

[4] We use the standard cropped version available at `www.cs.uiuc.edu/homes/dengcai2/Data/data.html`. Each image has size 64×64 pixels before illumination compensation, the feature extraction is performed on a downsampled (32×32) version of the illumination-compensated images.

Norm	Rec, rate
ℓ^2 unweighted	86.3%
ℓ^2 IDF-weighted	86.7%
ℓ^1 unweighted	89.3%
ℓ^1 **IDF-weighted**	**89.4%**

Table 1. Recognition rate for various classifier norms.

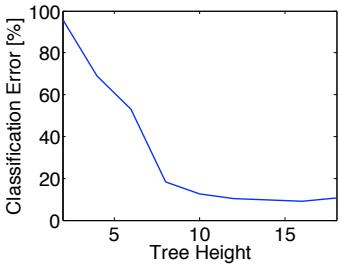

Figure 5. Classification error vs. tree height for the PIE database.

While this dataset has relatively few subjects, its small size allows us to extensively investigate the effect of various algorithm parameters. Moreover, the variability present in the database, due to moderate pose and expression, is a good proxy for the conditions our algorithm is designed to handle.

Histogram distance metric. We consider four distance metrics between histograms, corresponding to the ℓ^1 and ℓ^2 norms, with and without IDF weighting. Table 1 gives the recognition rate in this scenario. In this example, the IDF-weighted versions of the norms always slightly outperform the unweighted versions, and ℓ^1 is clearly better than ℓ^2. Based on its good performance here, we adopt the IDF-weighted ℓ^1 norm for the rest of our experiments.

Tree depth. We next investigate the appropriate tree height h for recognition. Motivated by the result of the previous experiment, use the IDF-weighted ℓ^1-norm as a histogram distance measure. We again use the PIE database, and induce a single randomized tree. We compare the effect of binning at different levels of the tree. Figure 5 plots the misclassification error as a function of height. Notice that the error initially decreases monotonically, with a fairly large stable region from heights 8 to 18. The minimum error, 9.2%, occurs at $h = 16$.

Forest size. We next fix the height h, and vary the number of trees in the forest, from $k = 1$ to $k = 15$. Table 2 gives the recognition rates for this range of k. While performance is already quite good (89.4%) with $k = 1$, it improves with increasing k, due to the smoothing effect seen in Figure 4. As the time and space complexity of our algorithm is linear in the size of the forest, even larger k may be practical for some problems. Here, though, we fix $k = 10$, to keep our online computation time less than 1 second per image.

Forest size	1	5	10	15
Rec. rate	89.4%	92.4%	93.1%	93.6%

Table 2. Recognition rate vs number of trees.

6.2. Large-scale recognition experiments

Based on the observations from the previous section, we next perform a series of increasingly challenging large-scale recognition experiments. To reduce the risk of overfitting each individual dataset, we fix the tree parameters as follows: the number of trees in the forest is $k = 10$. Recognition is performed at depth 16, using the IDF-weighted ℓ^1-distance between histograms.

Standard datasets. We test our algorithm on a number of public datasets. The first, the ORL database [21] contains 400 images of 40 subjects, taken with varying pose and expression. We partition the dataset by randomly choosing 5 images per subject as training and the rest as testing. The next, the Extended Yale B database [8], mostly tests illumination robustness of face recognition algorithms. This dataset contains 38 subjects, with 64 frontal images per subject take with strong directional illuminations. For this dataset, we use a random subset of 20 images per subject as training and the rest as testing. We also again test on CMU PIE [23], with the same random partition described in the above experiments.

Finally, we test on the challenging Multi-PIE database [10]. This dataset consists of images of 337 subjects at a number of controlled poses, illuminations, and expressions, taken over 4 sessions. Of these, we select a subset of 250 subjects present in Session 1. We use images from all expressions, poses `04_1,05_0,05_1,13_0,14_0`, and illuminations `4,8,10`. We use the Session 1 images as training, and Sessions 2-4 as testing. We apply the detection and geometric rectification stages of our algorithm to all $30,054$ images in this set. The rectified images are used as input both to the remainder of our pipeline and to the other standard algorithms we compare against.

To facilitate comparison against standard baselines, for the first three datasets we use standard, rectified versions[5]. For MultiPIE, no such rectification is available. Here, we instead run our full pipeline, from face and eye detection to classification. For comparison purposes, the output of the geometric normalization is fed into each algorithm. In addition to being far more extensive than the other datasets considered, MultiPIE provides a more realistic setting for our algorithm (and its competitors), in which it must cope with real misalignment due to imprecise face and eye localization.

Table 3 presents the result of our algorithm, as well as several standard baselines (PCA, LDA, LPP), based on linear projection. As expected, our method significantly outperforms these baseline algorithms. Moreover, the performance approaches the best reported on these splits (e.g., 97.0% for ORL and 94.6% for PIE, both with orthogonal rank one projections [12], and 94.3% for Ext. Yale B with orthogonal LPP [4]). For the newer MultiPIE dataset, our system performs over twice as well as baseline algorithms. This is not surprising, since these algorithms have no intrinsic mechanism for coping with misalignment[6]. The overall recognition rate of all the algorithms is lower on MultiPIE though, confirming the challenging nature of this dataset.

	ORL	Ext. Yale B	PIE	MultiPIE
PCA	88.1%	65.4%	62.1%	32.6%
LDA	93.9%	81.3%	89.1%	37.0%
LPP	93.7%	86.4%	89.2%	21.9%
This work	**96.5%**	**91.4%**	**94.3%**	**67.6%**

Table 3. Recognition rates across various datasets.

Uncontrolled data: Labeled faces in the wild. While the above results are encouraging, performance on such well-controlled datasets is not necessarily indicative of good performance in real web applications such as image search and image tagging. We therefore further test our algorithm on the more challenging Labeled Faces in the Wild dataset [13]. This database contains 13,233 uncontrolled images of 5,749 public figures collected from the internet.

To facilitate comparison with the state of the art, we follow the training and testing procedure suggested in [13]. Here, rather than recognition, the goal is to determine if a given pair of test faces belong to the same subject. We therefore dispense with the nearest-histogram classification step, and simply record the IDF-weighted ℓ^1 distance between each pair of test histograms. Different thresholds on this distance give different tradeoffs between true positive rate and false positive rate, summarized in the receiver operating characteristic (ROC) curve in Figure 6. In this setting, our algorithm achieves an equal error rate of 32%. This significantly surpasses baseline algorithms such as PCA [25], and approaches the performance of more sophisticated algorithms in the low false-positive-rate regime. One additional advantage of our algorithm is the weak dependence on the training data. In particular, we can obtain similar performance using randomized trees trained on completely different datasets. We demonstrate this using the PIE database as training and the LFW as testing. Figure 6 plots the result. In this scenario, performance actually improves: the equal error rate decreases to 28%, the ROC strictly dominates that generated by training on the LFW data itself. The performance equals that of supervised methods such as [20] (denoted `Nowak` in Figure 6), and falls within of the current best result on this data,

[5] `www.cs.uiuc.edu/homes/dengcai2/Data/data.html`

[6] Although LPP can adapt to nonlinear structure in the data.

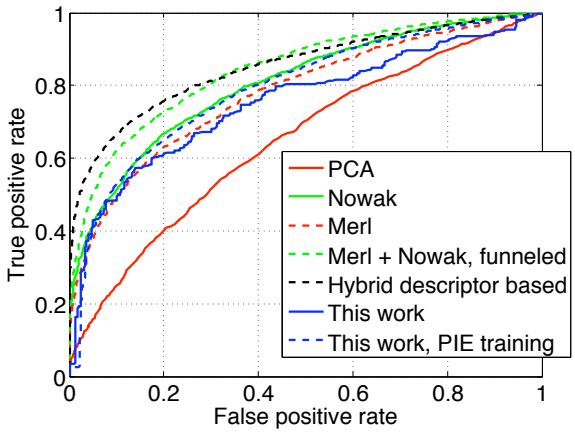

Figure 6. Receiver Operating Characteristic for Labeled Faces in the Wild. "Nowak" refers to [20]. "Hybrid descriptor based" refers to [30].

	PIE → ORL	ORL → PIE	PIE → MultiPIE
PCA	85.0%	55.7%	26.5%
LDA	58.5%	72.8%	8.5%
LPP	17.0%	69.1%	17.1%
This work	**92.5%**	**89.7%**	**67.2%**

Table 4. Recognition rates for transfer across datasets.

due to [30] (denoted `Hybrid descriptor based`; for a description of the remaining methods, please see `vis-www.cs.umass.edu/lfw/results.html`).

Generalization across datasets. One advantage of using a weakly supervised or even random classification scheme is that it provides some protection against overfitting. We demonstrate this advantage quantitatively by training on one dataset and then testing on completely different datasets. Methods which are prone to overfitting are likely to fail here. Table 4 reports the recognition rates for several combinations of training and test database. Comparing to Table 3, notice that our algorithm's performance decreases less than 5% when trained and tested on completely different datasets. The performance of PCA degrades similarly, but remains substantially lower. The performance of more complicated, supervised algorithms such as LDA and LPP drops much more significantly. For example, when trained on ORL and tested on ORL, LDA achieves a 94% recognition rate, which drops to 58% when trained on PIE and tested on ORL.

7. Extensions and Some Remarks

The approach outlined above can be extended and modified in several ways. First, if the number of training examples per subject is large, rather than retaining one histogram per subject it may instead be appropriate to retain a single histogram per class. We find that this degrades performance only moderately, for example, reducing performance on the ORL database from 96.5% to 92.5%.

In would also be interesting to investigate other classifiers besides nearest neighbor for the histogram matching step. For example, as is popular in histogram-based image categorization, one could learn a support vector machine classifier in the histogram space. Simple linear classifiers such as LDA or supervised LPP could also be applied[7] to the histogram, effectively treating the quantization process as a feature extraction step.

The proposed approach demonstrated superb performance in our experiments, especially when training and testing are performed on distinct datasets. Here we summarize some of the key observations obtained from our experiments, as well as our best interpretation of them.

1. We have seen that the recognition rate tends to increase as the height of the forest increases. This naturally raises questions about overfitting with excessively tall trees. While we have not observed this, we *have* observed that for transferring between databases, recognition performance can be improved by considering the top L levels of the tree (say, $L = 10$). Thus overfitting is a much larger problem in transfer experiments than in recognition experiments. This suggests that the top L levels of the tree actually adapt to structures that are common to all human faces, while the remaining (lower) levels fit much more specific aspects of the training database.

2. In all examples we have tried, increasing the number of trees improves (or at least does not decrease) recognition performance. Figure 4 suggests that this may be at least partially because aggregating the spatial boundaries of the bins produces a shape that is much more tightly tuned to the type of patch being quantized (e.g., eye corners). If the performance is indeed guaranteed to improve with more trees, it is interesting to ask if there is any sense in which the quantization regions or soft similarities are converging. If the limiting shapes have simple forms, this might lead to even faster classifiers with equally good performance.

3. In experiments with the Extended Yale B database, which explicitly tests illumination robustness, we find that removing the self-quotient normalization step reduces the recognition rate by over 9%, from 91.4% in Table 3 to 83.2%. Nevertheless, it may be that for less extreme illuminations present in real-world images, some invariance is already conferred by the feature descriptor itself. In the other direction, it would be interesting to better understand when one can get away

[7] In limited trials, we did not see significant improvement with this approach, suggesting that the histogram distance metric used here is already quite appropriate for recognition.

with simple image-based rectification, and when more complicated illumination models are required.

4. We have argued that forming random projection trees in the expanded (feature + coordinate) space yields a spatially varying implicit matching scheme. Our visualized examples and good recognition performance give indirect evidence that this is indeed the case.

8. Conclusion

We have introduced a new approach to face recognition in semi-constrained environments, based on at implicit matching of spatial and feature information. The proposed method performs competitively with existing linear projection approaches. Because of its weakly supervised nature, it also performs well in transfer tasks across datasets.

References

[1] R. Baraniuk and M. Wakin. Random projections of smooth manifolds. *Foundations of Computational Mathematics*, 2008.

[2] P. Belhumeur, J. Hespanda, and D. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9), 2003.

[4] D. Cai, X. He, J. Han, and H.-J. Zhang. Orthogonal laplacianfaces for recognition. *IEEE Transactions on Image Processing*, 2006.

[5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[6] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *Proc. ACM Symposium on Theory of Computing*, 2008.

[7] Y. Freund, S. Dasgupta, M. Kabra, and N. Verma. Learning the structure of manifolds using random projections. In *Proc. Neural Information Processing Systems*, 2007.

[8] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[9] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093, 2006.

[10] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. In *Proc. IEEE Conference on Face and Gesture Recognition*, 2008.

[11] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using Laplacianfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[12] G. Hua, P. Viola, and S. Drucker. Face recognition using discriminatively trained orthogonal rank one tensor projections. In *Proc. CVPR*, 2007.

[13] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical Report 07-49, 2007.

[14] W. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a Hilbert space. In *Conf. on Modern Analysis and Probability*, pages 189–206, 1984.

[15] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B. Lu. Person-specific SIFT features for face recognition. In *Proc. ICASSP*, volume 2, pages 593–596, 2007.

[18] Moosman, B. Triggs, and F. Jurie. Randomized clustering forests for building fast and. discriminative visual vocabularie. In *Proc. Neural Information Processing Systems*.

[19] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, 2006.

[20] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *Proc. CVPR*, 2007.

[21] F. Samaria and A. Harter. Parameterization of a stochastic model for human face identification. In *Proc. of IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, December 1994.

[22] C. Silpa-Anan and R. Hartley. Optimised KD-trees for fast image descriptor matching. In *Proc. CVPR*, pages 1–8, 2008.

[23] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination and expression database. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

[24] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *Proc. CVPR*, 2008.

[25] M. Turk and A. Pentland. Eigenfaces for recognition. In *Proc. CVPR*, 1991.

[26] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[27] H. Wang, S. Li, and Y. Wang. Generalized quotient image. In *Proc. CVPR*, pages 498–505, 2004.

[28] S. Winder and M. Brown. Learning local image descriptors. In *Proc. CVPR*, pages 1–8, 2007.

[29] L. Wiskott, J. Fellous, N. Kuiger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7), 1997.

[30] L. Wolf and T. H. andY. Taigman. Descriptor based methods in the wild. In *Proc. Faces in Real-Live Images Workshop, European Conference on Computer Vision*, 2008.

[31] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Proc. CVPR*, 2008.