Speeding up 2D-Warping for Pose-Invariant Face Recognition

Harald Hanselmann and Hermann Ney
Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Germany
surname@cs.rwth-aachen.de

Abstract—Recently, state-of-the-art recognition accuracies for pose-invariant face recognition have been achieved by using 2D-Warping methods in a nearest-neighbor framework. However, the main drawback of these methods is the high computational complexity. In this paper we address this issue. We use a simple and fast method to get a rough estimate of a 2D-Warping. This estimate can then be used to apply an image dependent warprange on the 2D-Warping algorithm, limit the possible poses or preselect the most likely classes. By this method we are able significantly reduce the runtime of a recently proposed 2D-Warping algorithm without sacrificing recognition accuracy.

I. INTRODUCTION

Next to illumination, occlusion and facial expression, pose-variation is one of the main challenges in face recognition. The approaches to pose-invariant face recognition proposed in the literature range from subspace methods [16], [5], [23] over methods based on frontal face reconstruction and 3D-Models [3], [22], [28], [17] to algorithms using 2D-Warping [1], [21], [2], [11], [9]. For the latter, warpings (image matchings) between gallery (train) and test images are computed to generate a warping score that can be used in a nearest neighbor classifier [14]. These methods do not require a training step and can achieve state-of-the-art results with only one gallery image per person. In this context the work in [9] recently reported good results with a fully automatic recognition setup on the CMU-MultiPIE database, one of the most challenging benchmark databases for pose-invariant face recognition. Due to the high computational cost of generating the 2D-Warpings, the runtime is the main drawback of the method in [9]. One possible solution to this problem is parallelization. On one hand the warping algorithm can be implemented using parallelization frameworks such as NVIDIA CUDA [20], on the other hand the computation of the warpings can be distributed across many cores, because they are independent from each other. Another possibility is to reduce the runtime of the 2D-Warping method by adjusting the warping algorithm and classification process. In this paper we compute a fast estimate of the warping and use it to impose restrictions on the computationally expensive warping algorithm originally proposed in [9]. By this method, the latter can be forced not to compute warpings that are unlikely to lead to a successful classification and thus reduce the runtime. Additionally, the computed estimate can be used for pre-selection strategies.

The paper is structured as follows. In Section II we describe the face recognition pipeline used in [9] followed by a review of 2D-Warping in Section III. In Sections IV, V and VI we describe our proposed method. It is evaluated on the CMU-MultiPIE database in Section VIII and we finish with a conclusion and outlook in Section IX.

II. POSE-INARIANT FACE RECOGNITION PIPELINE

The authors in [9] use a face recognition pipeline that employs fully automatic classification given only a single gallery image per class. The test images are processed by a face detector [12] resulting in a rough bounding-box of the face. The gallery images are cropped and normalized manually. Both, test and gallery images are resized to a resolution of \(204 \times 256\) pixels and normalized against illumination changes [25]. To classify a given test image, the nearest neighbor rule with 2D-Warping scores as similarity measure is used. This means each gallery image has to be mapped to the test image to get a score for the corresponding class. To avoid occlusions caused by pose variations, the left and the right half of each gallery image is mapped independently as proposed in [1]. The one with the best score is selected for nearest neighbor classification. This is done for all test images, since at test time it is not known whether a pose-induced occlusion is present or not. As a result, the gallery images used for matching are half the size of the test images. An example of the matching between a halved gallery image and different test images is given in Fig. 1. It shows the left half of the gallery image and the test images after preprocessing and face detection. As a result of the preprocessing the images are roughly aligned with respect to the vertical axis but not the horizontal axis. Fig. 1 further shows the result of mapping the gallery image to the test images. It can be observed that the target region (everything covered by the deformed grid in the test image) is a patch of roughly the same size as the gallery image. However, the 2D-Warping algorithm searches the whole image for the optimal solution. This motivates to find a quick approximation of...
the target patch in advance and restrict the 2D-Warping algorithm accordingly.

III. 2D-WARPING

The goal in 2D-Warping (2D Image-matching) is to find the optimal mapping between two images regarding a specific cost function. In [21] the problem is defined as follows. Given a source image \( X \) with dimensions \( I \times J \) and a target image \( R \) with dimensions \( U \times V \) (in this paper we assume \( U = I \) and \( V = J \); the gallery image is the source image while the test image is the target image) one is interested in warping \( w \) that assigns a pixel \( w_{ij} = (u, v) \in R \) to each pixel \( (i, j) \in X \). This mapping is computed by optimizing the energy function

\[
E(X, R, \{w_{ij}\}) = \sum_{i,j} [d(x_{ij}, r_{w_{ij}}) + T(w_{i-1,j}, w_{ij}) + T(w_{i,j-1}, w_{ij})].
\]

In this equation function \( d(\cdot) \) denotes a distance function between two feature vectors \( x_{ij} \) and \( r_{w_{ij}} \), representing pixel \( (i, j) \) and \( w_{ij} \). The distance function can also be extended to include a small local context, e.g. instead of computing the distance between the feature vectors of two pixels directly the distances within a squared window around the pixels are accumulated and normalized by the window size. A cutoff-threshold \( \tau \) can be used on the local distances to be more robust against outliers. The relative penalty function \( T(\cdot) \) is the smoothness term. Different displacement vectors of neighboring pixels are penalized by using the Euclidean distance on the warping positions weighted by a factor \( \alpha \). Additionally monotonicity and continuity constraints are integrated [27].

Due to the 2D-dependencies, optimizing Formula (1) is NP-complete [15]. However, there are several algorithms available to find an approximative solution [14], [1], [21], [9]. Two of them are used in this work, namely Zero-Order Warping (ZOW) [14] and Two-Level Dynamic Programming with Lookahead (2LDP-LA) [9].

A. Zero-Order Warping

Zero-Order Warping (also referred to as Image Distortion Model) [14], [21] discards all 2D-dependencies and optimizes each pixel independently. As a result, no relative penalty function can be defined. To apply at least some kind of restriction on the warping, an absolute penalty \( T_\Delta(\cdot) \) is used:

\[
E(X, R, \{w_{ij}\}) = \sum_{i,j} [d(x_{ij}, r_{w_{ij}}) + T_\Delta((i, j), w_{ij})],
\]

where

\[
T_\Delta((i, j), w_{ij}) = \begin{cases} \alpha d_\text{pen}((i, j), w_{ij}) & \text{if } |v - j| \leq \Delta \land |u - i| \leq \Delta \\ \infty & \text{else} \end{cases}
\]

\( T_\Delta(\cdot) \) is based on the absolute displacement of pixel \((i, j)\) and penalizes deviations by using a distance function \( d_\text{pen}(\cdot) \). Additionally an absolute warprange \( \Delta \) is included as an upper bound for possible displacements. This approach is only promising if the the two images to be matched are already roughly aligned and the dimensions are normalized.

With a complexity of \( O((IJ)(2\Delta + 1)) \) ZOW is quite fast, but since neighboring pixels are mapped independently it can not capture image structures. Warprange, absolute penalty and local context can only compensate for that to a limited extend.

B. Two-Level Dynamic Programming

In contrast to ZOW, the Two-Level Dynamic Programming algorithm introduced in [9] does not need to discard the 2D-dependencies between the pixels. Instead, an approximative optimization of Formula (1) is done by sequentially optimizing the columns over a representative row of the 2D-grid, guided by a lookahead (2LDP-LA). This method has been applied to pose-invariant face recognition and achieved state-of-the-art results for a fully automatic recognition setup (see Section II). However, the computation is still complex and time consuming (the authors report 16 seconds for one warping). One reason for this is that the optimization is unrestricted in the sense that each pixel from the source image can be mapped to any position in the target image. This is a nice property in general, but unnecessary in the case of face recognition. For example, a warping where all pixels are mapped to the same target pixel is possible and considered by 2LDP-LA. But if this warping would be optimal, then it is very unlikely that this leads to a successful recognition. Therefore, we do not need to explore this option in the first place. This can be achieved by introducing a warprange as in ZOW [14]. However, as in ZOW, an absolute warprange leads to the problem that the images have to be roughly aligned, which is not the case in general. For this reason we use an image dependent warprange which is described in the next section.

IV. IMAGE DEPENDENT WARPRANGE

Using a warprange to reduce the search space for possible warping mappings is often used, e.g. for ZOW [14], Tree-Serial Dynamic Programming [19] or the maximal disparity in stereo vision [24]. As mentioned before, applying an absolute warprange requires the images to be roughly aligned. If pixel \((i, j)\) is forced to be mapped to the region \((i \pm \Delta, j \pm \Delta)\) for a small \( \Delta \), then a larger translation makes reaching the optimal target patch impossible. For this reason, we use a warprange that depends on the source and target image at hand. This is done by altering Formula (1) to include an additional penalty function defining the warprange \( \delta \)

\[
E(X, R, \{w_{ij}\}) = \sum_{i,j} [d(x_{ij}, r_{w_{ij}}) + T(w_{i-1,j}, w_{ij}) + T(w_{i,j-1}, w_{ij}) + T_\delta((i, j), w_{ij})],
\]

where
to get a rough estimate of the right target region for image images to be matched (X offset for the warprange. They are dependent on the two dimensions. The variables component to have separate warpranges for both image be used.

ZOW (described in the next section) or facial landmarks can (for details see [9]), reducing this to (2

heavily depends on the search space UV introduced in the following.

offsets for 2LDP-LA-W we consider two options, which are to compute a smooth warping, but 2LDP-LA-W on the other hand generates a better result due to the forced smoothness by the warprange.

V. TARGET PATCH APPROXIMATION

To approximate the target patch and determine the relative offsets for 2LDP-LA-W we consider two options, which are introduced in the following.

A. Sliding Window ZOW

To apply ZOW when the images are not aligned is problematic, because on one hand a large warprange must be chosen to compensate for large translations but on the other hand a large warprange leads to non-smooth warpings which lead to bad classification results. To overcome this problem we apply ZOW in a sliding window (SW-ZOW), similar to template matching (e.g. [10]). This is done in a pseudo-2D fashion using a two-pass procedure, since a complete search would be too computationally expensive. In the first pass, a horizontal sliding window is used while the vertical offset is fixed to zero. The horizontal offset that leads to the best match is then fixed and a vertical sliding window is used to refine the result. We do the horizontal pass first based on the assumption that the images are already roughly aligned in the vertical dimension (Section II). Note that the sliding window does not define a strict bounding box and pixels can be warped outside the window as far as the warprange ∆ allows.

The complexity of SW-ZOW is as follows. Assuming \( U \gg I \) and \( V \gg J \) the minimal horizontal offset for \( o_{h}^{X,R} \) is given by \(-\Delta\) and the maximum is \( U - I + \Delta\). Therefore the offset in the horizontal dimension can take a value in the range of \([-\Delta, ..., U - I + \Delta]\). For each of these offsets a ZOW-model has to be optimized. As mentioned in Section III-A the complexity for ZOW is \( O(IJ(2\Delta + 1))\), not considering boundary effects. As a result the complexity for the horizontal pass of the sliding window ZOW is \( O((U - I + 2\Delta + 1)IJ(2\Delta + 1))\). The same considerations apply for the vertical pass which means in total ZOW has to be applied \((U - I + V - J + 4\Delta + 2)IJ(2\Delta + 1))\). This complexity has to be added to 2LDP-LA-W as overhead, but the results in Section VIII show that overall the runtime is improved.

Applying ZOW in this manner can be interpreted as trying different offsets for 2LDP-LA but instead of computing expensive warpings with 2D-dependencies the dependencies are discarded to get a fast estimate of the quality of the offset. In other words, while 2LDP-LA searches the image while maintaining expensive 2D-dependencies, SW-ZOW searches the image without them.

B. Landmarks

The SW-ZOW is useful to find the most likely matching region when nothing is known about the test image. However, if facial landmarks are available (e.g. manual annotations or by applying a landmark detector such as [30]), they can be used directly to locate the best region in the test image. E.g. from Fig. 1 it can be observed that the landmarks eye and mouth are mapped onto each other. If the landmarks are available at test time, then the distances of the landmark positions between gallery and test image can be used to determine the offset for 2LDP-LA-W by taking the average for all landmarks. A fast estimate of the optimal 2D-Warping can then be obtained by applying ZOW once at the determined offset or applying SW-ZOW in a small sliding
window around the offset to weaken the effect of inaccurate landmarks.

A common preprocessing step for face recognition is to apply a pre-alignment procedure that uses landmarks and pose information to normalize rotation and scale of the faces (e.g., the work in [5]). Such techniques are also helpful for 2LDP-LA-W, since pre-aligned images allow the use of tighter warpranges.

VI. PRE-SELECTION

In the previous sections the goal was to reduce the complexity of a single 2LDP-LA warping. However, for the classification using 2LDP-LA in a nearest-neighbor framework, several such warprings have to be computed. Given $C$ classes and the face recognition pipeline from Section II, $2C$ different 2LDP-LA warprings are needed to classify an image (left and right half of each gallery image are matched to the test image). This number can be significantly reduced by pre-selection techniques similar to pre-filtering [26]. To be more precise, we use two pre-selection strategies.

First, we can decide early on whether the left or the right half of the gallery image is better suited for classification (LR pre-selection). In case the pose of the test image is known we warp the left half. In case we do not have any pose information we compute the SW-ZOW score for both halves and take the one with the lower score.

Second, we can use the SW-ZOW energies to eliminate unlikely classes by ranking the gallery images according to their SW-ZOW scores and keeping only the best $k$ for classification with 2LDP-LA (class pre-selection), similar to [8].

VII. TECHNICAL DETAILS

We extended the implementation used in [9] and [6] and follow the proposition in [29] by using integral images to compute the local contexts of the distance function. Additionally, we used the NVIDIA CUDA framework [20] to implement a parallel version of the 2LDP-LA warping algorithm. The runtime measurements presented in Section VIII were performed using a notebook with a 2.80 GHz Intel Core i7-3840QM CPU and a NVIDIA Quadro K2000M GPU averaging over ten representative warprings, while the recognition results were obtained using a cluster with over 100 AMD CPUs. For the latter we observed significantly higher runtimes for single warprings than on the notebook, but comparable speedup factors.

VIII. EXPERIMENTAL EVALUATION

We evaluate the proposed method on the CMU-MultiPIE [7] database, which is a popular benchmark for pose-invariant face recognition. The database has 337 classes and the images were recorded over four different recording sessions. Amongst others, the database offers variations in pose from $-90^\circ$ to $+90^\circ$ yaw (13 different poses in total). Additionally there are two poses (08_1 and 19_1) where next to $45^\circ$ yaw also a slight variation in pitch is present to simulate the typical angle of a surveillance camera. In the original database the images for the two surveillance poses are delivered upside down, therefore we rotate them by $180^\circ$ prior to all our experimental setups.

A common approach for pose-invariant face recognition is to use the first 100 subjects from the first session for model training and the remaining 149 subjects from the same session for testing. The illumination is constant in all images (flash 07) and pose 05_1 is used as gallery image (one image per subject), while the other 14 poses are used for testing (overall 2086 test images). Since our classification approach does not involve a training step, we use the first 100 subjects (specifically pose 11_0 ($-90^\circ$)) to tune our basic warping parameters: The warping penalty $\alpha$, the distance threshold $\tau$ and the size of the local context. As in [9] we use PCA reduced U-SIFT features [18], [13], [4].

We use two basic setups in our experiments.

1) Face detection setup: First we use the same pipeline as in [9], which has already been described in Section II. In this case no manual annotations are used for the test image and the recognition is done fully automatic. Only the gallery images are normalized using the annotations and alignment procedure from [5]. This setup is well suited to evaluate the performance of 2LDP-LA-W in combination with SW-ZOW, but since no landmarks are available we cannot test 2LDP-LA-W in combination with facial landmarks.

2) Landmark setup: To evaluate 2LDP-LA-W with landmarks to localize the target region we use a second setup where facial landmarks are given for all images. Additionally we require the yaw and pitch angles to be known in order to perform the alignment proposed in [5] also for the test images. For this purpose we use the manual annotations from [5] to see the full potential of our method. However, as a result, the experiments for this setup are not fully automatic. Thus we repeat the experiments using a publicly available landmark detector and pose estimator [30]. For simplicity we use one of the pre-trained models supplied by the authors. Running a landmark detector and pose estimator causes additional computation time but has to be done only once for each test image and not for each test and gallery image pair.

It should be noted that we only compare the runtime of the matching algorithms in this section. The additional steps (face detection / landmark detection, preprocessing, feature extraction etc.) are not considered.

A. Face detection setup

Table I shows the results for 2LDP-LA-W with SW-ZOW and different values for the warprange of SW-ZOW $\Delta$. Additionally we give the runtime and speedup factor compared to the baseline 2LDP-LA for one image matching. The runtimes for 2LDP-LA-W are given including the runtime for SW-ZOW. The latter can also be used directly for classification.
and the accuracy of SW-ZOW is remarkable considering it is a much simpler and faster warping method. However, adding 2LDP-LA-W gives a classification performance boost of almost 10% absolute. For the different values of $\Delta$ (the warpranges $\delta_h$ and $\delta_v$ are fixed to 20 and 15) the recognition accuracy of 2LDP-LA-W is around 86% and very close to the baseline accuracy of 2LDP-LA.

As a compromise between accuracy and speed we select $\Delta = 6$ and do further experiments with different values for $\delta_h$ and $\delta_v$. The results are shown in Table II. Also for these experiments the recognition accuracy is around 86%. For $\delta_h = 15$ and $\delta_v = 15$ we even achieve the same accuracy as the baseline, although almost four times faster. Note that applying only SW-ZOW with $\Delta = 6$ leads to a speedup factor of 9.3. This defines an upper bound for the speedup that can be achieved adding 2LDP-LA-W, since SW-ZOW still has to be applied for all gallery images.

Table III gives results for the pre-selection of the left or right half of the gallery image based on SW-ZOW scores (see Section VI). While the recognition accuracy drops by 0.3% absolute, the speedup factor compared to 2LDP-LA is now at 5.4 (based on the runtimes in Table II).

**B. Landmark setup**

In Table IV the results with landmarks are given. We only use the landmarks of the eyes to find the optimal target patch since preliminary experiments have shown this to be optimal. Note that for the results in Table IV we already applied LR pre-selection based on the sign of the pose angles (see Section VI) for all methods including the baseline 2LDP-LA. It is no surprise that with manual annotations and pre-alignment the baseline accuracy is improved to 90.5%. For 2LDP-LA-W we keep $\Delta = 6$, but use lower values for $\delta_h$ and $\delta_v$, since the images are pre-aligned. We set the parameters to 10 and 6, respectively. Using 2LDP-LA-W with full SW-ZOW as in the previous section already leads to a small improvement of the recognition accuracy compared to 2LDP-LA while being six times faster. Completely skipping SW-ZOW and using only the landmark information to approximate the target patch for 2LDP-LA-W leads to another increase in accuracy and this setup is even 20 times faster. These two results indicate that on one hand, restricting 2LDP-LA by a warprange leads to better classification results and on the other hand SW-ZOW does not always find the optimal target patch.

For the automatic landmark detector we use SW-ZOW in a window of 13 pixels around the detected landmark explaining the increased runtime compared to manual landmarks. With this setting, 2LDP-LA-W reaches an accuracy of 88.8%, which is not only better than the best result in an automatic setup so far (see Table II), but also very close to the results achieved with manual annotations. In fact, in Table V we can see that there is only a large difference for the surveillance results, which is due to a high number of errors by the landmark detector (the pre-trained models [30] were not trained on images with tilt).

**C. Class pre-selection**

In Fig. 3 the effect of pre-selecting the best $k$ classes using SW-ZOW scores on the recognition accuracy is illustrated. Results are given for both experimental setups and all results are after the respective LR pre-selection. For the face detection setup with $k = 149$ the accuracy corresponds to the result in Table III. When decreasing $k$ the accuracy is constant at around 86% until with $k = 36$ the accuracy drops by 1% absolute compared to 2LDP-LA.

The overall speedup factor at this point is 7.8 (SW-ZOW is still done
D. Parallel implementation

For the previous experiments the runtimes were computed using one CPU core, but the used warping algorithm is also parallelizable. To get an idea of how much this can improve the runtime we compare the CPU implementation with a parallel GPU implementation using the NVIDIA CUDA framework. Where a full 2LDP-LA warping took 8.88 seconds on the CPU, we measured 1.37 seconds on the GPU. Similarly the runtime for 2LDP-LA-W in combination SW-ZOW ($\Delta = 6$, $\delta_h = 15$ and $\delta_v = 15$) was improved from 2.31 seconds to 0.6 seconds. Note that due to numerical differences between platforms and implementations (e.g. double vs. single precision floating point numbers) the computed energies differ and can therefore also alter the classification results.

E. Comparison to the State-of-the-Art

In Table V (and more detailed in Fig. 4 and 5) we compare the recognition performance of 2LDP-LA-W with some state-of-the-art methods that use a similar setup. We differentiate between methods using manual landmarks and methods using a fully automatic testing procedure and group the images into three sets, near-frontal and near-profile with six images each, and the two surveillance images. While many methods reach perfect or close to perfect performance on the near-frontal images, the near-profile and surveillance images are more difficult. To the best of our knowledge the maximum likelihood correspondence estimation (MLCE) [17] reports the best accuracy for the near-profile images, but the authors give no results for the surveillance images. MLCE as well as the partial least squares method (PLS) from [5] achieve better results on the near-profile images than our method, but it should be noted that MLCE and PLS both train pose specific models. This is in contrast to our method since we use only one parameter set for all poses in order to be more robust against inaccurate landmark and pose information. However, it is also possible to optimize pose-specific parameter sets to improve the accuracy.

To the best of our knowledge, 2LDP-LA-W in combination with automatic landmarks achieves with 88.8% accuracy the best performance for methods with a fully automatic recognition setup. This is also remarkably close to the approaches using manual annotations.
TABLE V

<table>
<thead>
<tr>
<th>Method</th>
<th>Manual test image annotations</th>
<th>Near-frontal[%]</th>
<th>Near-profile[%]</th>
<th>Surveillance[%]</th>
<th>Total without surveillance[%]</th>
<th>Total[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2LDP-LA-9</td>
<td>no</td>
<td>99.78</td>
<td>69.69</td>
<td>98.70</td>
<td>84.73</td>
<td>86.72</td>
</tr>
<tr>
<td>2LDP-LA-W (Face detector)</td>
<td>no</td>
<td>99.44</td>
<td>70.13</td>
<td>98.30</td>
<td>84.79</td>
<td>86.72</td>
</tr>
<tr>
<td>2LDP-LA-W (Landmark detector)</td>
<td>no</td>
<td>100.00</td>
<td>80.43</td>
<td>80.54</td>
<td>90.21</td>
<td>88.83</td>
</tr>
<tr>
<td>2LDP-LA-9</td>
<td>yes</td>
<td>100.00</td>
<td>78.41</td>
<td>98.32</td>
<td>89.21</td>
<td>90.31</td>
</tr>
<tr>
<td>2LDP-LA-W</td>
<td>yes</td>
<td>100.00</td>
<td>80.43</td>
<td>98.99</td>
<td>90.21</td>
<td>91.47</td>
</tr>
<tr>
<td>PLS [57]</td>
<td>yes</td>
<td>98.88</td>
<td>82.10</td>
<td>97.60</td>
<td>90.30</td>
<td>90.07</td>
</tr>
<tr>
<td>MLCE [17]</td>
<td>yes</td>
<td>100.00</td>
<td>84.22</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

* Different illumination (flash 00).

IX. CONCLUSION

In this paper we have integrated an image dependent warprange to the 2D-Warping algorithm 2LDP-LA. This warprange is found by either applying ZOW in a sliding window, facial landmarks or a combination of both. The experiments on a pose subset of the CMU-MultiPIE have shown that by this approach, the runtime can be significantly improved by factors ranging from 3.5 to 29 for various setups without sacrificing accuracy (even higher when we tolerate slightly lower accuracies). On the contrary, we are able to improve the baseline accuracy for a setup with manual annotations as well as a fully automatic setup including a landmark detection and pose estimation step. To the best of our knowledge, the result of 88.8% accuracy is the best result achieved so far in a fully automatic setup.

While SW-ZOW is much faster than 2LDP-LA, there is still room for improvement. For example the step-size for the sliding window could be adjusted or different scales for the window could be used. The latter would make the method more interesting for other use cases than the face recognition pipeline used in this work.

REFERENCES