# Exploiting Perception for Face Analysis: Image Abstraction for Head Pose Estimation

Anant Vidur Puri and Brejesh Lall

Indian Institute of Technology, New Delhi, India
http://www.ee.iitd.ac.in

**Abstract.** We present an algorithm to estimate the pose of a human head from a single, low resolution image in real time. It builds on the fundamentals of human perception i.e. abstracting the relevant details from visual cues. Most images contain far too many cues than what are required for estimating human head pose. Thus, we use non-photorealistic rendering to eliminate irrelevant details like expressions from the picture and accentuate facial features critical to estimating head pose. The maximum likelihood pose range is then estimated by training a classifier on scaled down abstracted images. The results are extremely encouraging especially when compared with other recent methods.Moreover the algorithm is robust to illumination, expression, identity and resolution.

**Keywords:** Face, Head Pose, Non Photorealistic Rendering, Abstraction.

## 1 Introduction

Head pose estimation is an intriguing and actively addressed problem in computer vision[1]. The reason for this is the application potential of an accurate pose estimation system in Human Computer Interaction, which is one of the most upcoming research areas in recent times. Some of the applications in this field are emotion recognition, unobtrusive customer feedback, biological pose correction, and interactive gaze interfaces. Knowledge of head pose is also extremely useful in a host of other head and face related computer vision applications including surveillance and avatar animation. More recently, a lot of research has been done on facial expression recognition which is highly simplified if head pose is known apriori. Recognition is also benefitted by prior knowledge of head pose.

In view of all this, highly accurate systems have been proposed, including 3D models[2], fiducial point fitting[3], and machine learning (ML)[4] [5] techniques. But, the major challenge that thee methods face is pose estimation in the presence of extreme facial expressions. This is a pertinent problem as in any real-life setting, the human-face is seldom without any facial expressions. Also, a critical use of pose is in facial expression recognition. So any solution which is not reasonably robust to expression is futile. Most of these methods also suffer from one or more drawbacks such as high computational complexity[3], requirements for huge training sets[5], tedious alignment issues[4] [5], sensitivity to illumination[5] [2], personalized initialization, and non-scalability to estimate

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part II, LNCS 7584, pp. 319–329, 2012.

pose for multiple subjects[3] [2]. It is thus with an aim to simplify the problem and to seek an easy, fast, and accurate solution that we propose our algorithm. Our approach is essentially to seek appropriate features that can be leveraged by statistical learning techniques. The unique part is that it takes advantage of the fact that most images have a level of detail that is much more than what is required to estimate pose. So, we abstract the image using non-photorealistic rendering (NPR)[5]. NPR reduces the complexity of analysis while sending the information across in a better and easier form. This approach leads to a computationally light system suitable for real time use and scalable to multiple subjects. Abstraction affords robustness to identity and facial expressions. Also, our approach is relatively insensitive to skin and lighting variations. The obtained features lead to considerably fewer alignment issues than other ML techniques for head pose estimation. The method's accuracy and computational lightness make it ideal as a first step to estimate finer head pose. Fig. 1 illustrates the excellent performance of our algorithm showing the results on certain random images from the Internet.


**Fig. 1.** Successful examples

The rest of the paper is organized as follows. Section 2 describes image abstraction while section 3 describes the NPR algorithms used in our system. Sections 4 and 5 focus on the training and testing pipelines, respectively. We present our results in section 6, where we also describe a heuristic to estimate a usable output range of pose. The final section is the conclusion describing future scope and possible modifications to the algorithm.

## 2 Image Abstraction

The reason behind abstraction can be best understood through a series of examples shown in Fig.2. It shows the actual image of a person followed by the same image after it has undergone different forms of abstraction. These include segmentation of the color regions, identification of prominent contours, isolation of the skin region or viewing only very few features of the face like hairline and eyes.

**Fig. 2.** Various forms of image abstraction (left to right): original image, color segmentation image, contours image, skin detector output, cropped upper face

The thing common to all these images is that they contain far less detail (information) than the original image. While one has lesser colors, another has no colors but only strokes and one even has parts of the face missing. Yet they can all be used to infer pose. The crucial fact here is that our brain seldom needs so much detail to estimate pose. Similar claims of the brain using limited information from an image have been verified in works on perception and imaging. [6] shows how people can interpret shapes from line drawings just as well as they can from shaded images. Experiments have been conducted which show that humans take lesser response time to analyze a scene when shown cartoon drawings rather than real pictures [7]. Cavanagh [8] proposes that though we think we are seeing a complete person, we actually just see a line diagram of the person i.e. Fig.3. This brings us to the next important fact that even though our eyes see the complete image our brain uses only what is bare minimum required to perform the given task (in this case - estimate pose). A machine lacks this ability to filter out unnecessary details. However, if we can do the abstraction prior to analysis by a learning algorithm, it is possible to increase the efficiency and overall performance of many such algorithms. The lesser a machine has to learn the less confused it will get and the easier it will be to obtain results.

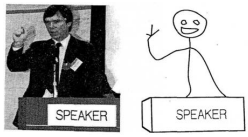


**Fig. 3.** What we see(left) and what we think we see

Abstraction, however, is a subjective term and can be taken to have many meanings. In our system, we achieve abstraction using non-photorealistic rendering techniques. In the next section, we describe NPR and how it leads to image abstraction.

## 3 Non-photorealistic Rendering

As the name suggests, non-photorealistic rendering of images seeks to give an artistic impression to images. This may be by simulating paintbrush strokes, creating segments of uniform colour, introducing a pen and ink illustration effect etc. Stylizing the image to make it look like a painting has an effect akin to that of abstraction i.e. to preserve necessary information while doing away with unnecessary details. The earliest well-documented experiments in this area were conducted as long as half a century ago [14]. These showed the clear advantage of non-photo realistically rendered images over real life pictures as far as speed of perception is concerned. [9] also presents a perceptual experiment that illustrates the mutual beneficence of NPR and perception. It also states, 'Artists have long realized that complex scenes, and even abstract ideas or concepts can often be visualized more effectively with simple images that hide much of the complexities of the physical interaction between light and matter[10]. Indeed, much of the work in non-photorealistic rendering is motivated by exactly this potential for effective visual communication [11].' This same fact can also be inferred from a simple observation: *most real life paintings lack extensive detail yet manage to draw attention to key features.* Paintings like the Mona Lisa (Da Vinci) or

The Potato Eaters (van Gogh) have not been painted with intricate detail yet they draw attention to salient features better than even high resolution images. The style and organization is such that they are rendered with meaningful abstraction to achieve more effective visual communication. Another notable example is the poster of Moulin Rouge by Toulouse-Lautrec (Fig. 4), which draws the viewer's attention to meaningful parts through abstraction (Refer [12] for a discussion on this work).


**Fig. 4.** Henri de Toulouse-Lautrec's 'Moulin Rouge'

NPR can thus reduce complexity of a computer vision algorithm by accentuating salient information. To investigate this prospect, we use two simple forms of NPR to simplify and abstract features of the head so as to make pose estimation easier. These are:

1) Regional Segmentation
2) Stroke Rendering

Both methods are explained in the following subsections.

### 3.1 Regional Segmentation

The method is essentially color based segmentation: seed growing followed by refinement steps. First, a seed is planted, and a region is grown by adding all neighborhood pixels with RGB values within a certain threshold. The algorithm then recursively calls itself on the neighbors of the last added pixel(s). Whenever the threshold is violated, a new seed is planted, and a new region is grown. This continues until every pixel is a part of some segment. This process is illustrated through Fig. 5.

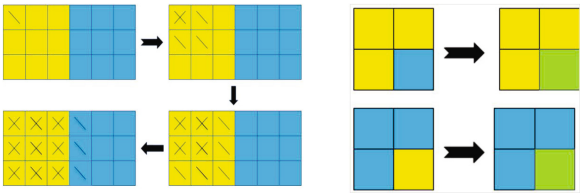

**Fig. 5.** (Left: Region growing using initial seed. The single line denotes pixels being processed in current step. Crossed pixels have been analyzed and are part of current segment. Right:Refining segments by smoothing out edges.

Refinement of segments happens in two steps. The first step involves smoothing the segment boundaries. This is done by considering four pixels of the initially segmented image at a time. If three out of four pixels belong to one segment then

all four are merged into the same segment by filling the fourth pixel with an intensity value that is intermediate between that pixel and the others, as shown in Fig. 5.

The second refinement step is repeated segmentation, i.e., segmentation and refinement are done repeatedly until the number of segments becomes constant. This ensures that the image cannot now be divided into more segments with that threshold value.

Fig. 7 shows similar results for images from the Pointing Database [13] (which is used for training and testing our pose estimation system). As can be seen, unnecessary irregularities in the image are smoothed out and important patches emerge.

We undertake two steps to make the system robust to illumination and skin color. First, we have used histogram equalization on the original images. Further, we normalize the pixel values of the NPR image i.e. $p_i = (p_i - \mu)/\sigma$, where $\mu$ and $\sigma$ are the mean and standard deviation of that NPR image.

It is noticed in the results, that, any facial contortions or extreme expressions are smoothed out and only blobs remain. Hairlines show up in all cases; eyes also result in distinct blobs. Often these two alone are sufficient to determine pose. The algorithm also detects mouth and nose as distinct blobs. Moreover, hair and skin can show up as different segments. Thus, the approach is adaptable to a wide range of people, environmental conditions, and facial expressions.

These observations are strengthened by the mean and variance images, shown in Fig. 8 (derived during training).The NPR images are converted to gray scale and used by the system during training and testing. The locally uniform regions in the mean images show that the images are smoothened to uniform color segments. We also get some regions of very low variance, which implies that critical features of the face are strongly captured. Often, shadows are visible on the face, and this may initially seem like a deterrent. On observing the learned model, however, we realize that the shadows and where they lie (which side below the chin, etc) are themselves cues for pose and are learned by the system. *Since only the basic features of the face, which remain more or less same throughout the dataset, are retained, the algorithm learns lesser but more relevant information and is thus able to respond better.*

### 3.2 Stroke Rendering

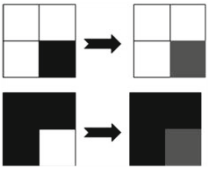

**Fig. 6.** Edge Refinement

The second method involves rendering the image using strokes like in line drawings. First, edges are detected using simple Sobel masks of size 3X3. These edges are then thresholded to retain only those edges which define major segments in the image. These edges are then refined by considering four pixels at a time; if three of them are black (i.e., they form an edge), the fourth one is given half the maximum grayscale value. Similarly if only one is black, it is reassigned half the maximum grayscale value. This is illustrated in Fig. 6.

The edges detected by using this method are, however, very thin. It could be difficult to train a system on such thin edges as the amount of data would be too little. So we do a thickening operation on the edges, similar to dilation. The mean and variance images for stroke rendering are shown in Fig. 8.

**Fig. 7.** NPR results for Pointing Database (top to bottom): original image, region segmented image, stroke rendered image

While the mean images do show some clarity in terms of making out the face and pose, there are very few smooth, low variance regions. This could challenge a system to learn a suitable pattern. However, the images become better as the pan angle becomes more acute. As far as computational performance is concerned, a Python implementation is currently able to non-photorealistically render at 3 fps (for both region segmentation and strokes) with image size 96X96 pixels while running on a 2.4 GHz Intel 3 Processor.

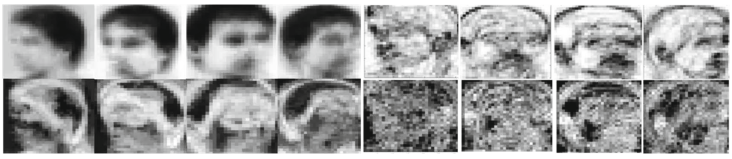

**Fig. 8.** Mean (top) and variance (bottom) images for different poses for regional segmentation(left) and stroke rendering(right)

## 4 Training

A learning-based system is expected to perform well if trained on abstracted images of human heads. We have explored this idea using a naive Bayes classifier. As mentioned before, we have used images from the Pointing Database (30 different images are available for each pose) for training our system. The training procedure is explained below:

(i) Images are cropped according to the boundary of the head and the cropped images are histogram equalized to be robust to illumination effects.

(ii) A non-photorealistic rendering algorithm is applied to generate abstracted images. The same parameters are maintained during training and testing.

(iii) The result is then converted to a gray scale, 32X32 frame. This step helps in normalizing the locations of the abstract regions across various head shapes. It also saves computations during both training and testing.

(iv) A statistical model is learned (Section 5) at each pixel $i$ for a given pose textitj($p_{ij}$), by computing the mean $\mu_{ij}$ and standard deviation $\sigma_{ij}$ of pixel values $x_{ij}$ , as shown in equations (1) and (2). Here, each pixel is assumed to be

statistically independent, which makes the computation easy because the probabilities across all pixels can be just multiplied to obtain the total probability; at the same time, the learned model is accurate enough for pose estimation. Separate models are trained for both NPR techniques. Currently, the system learns models from frontal, 0°, to 90° in steps of 15° for pan right and left. Also, for certain pan angle ranges($< 60°$), models for tilt up 30° and tilt down 30° are learned.

$$\mu_{ij} = \Sigma x_{ij}/N \tag{1}$$

$$\sigma_{ij}^2 = \Sigma(x_{ij} - \mu_{ij})^2/N \tag{2}$$

## 5 Estimating Pose

In a test image, the head region is first cropped. To the best of our knowledge several machine learning techniques impose a strict requirement on the cropped face. For example, in Huang et al. (2011), the crop is according to the positions of fiducial points. Finding these facial features itself is difficult, leading to a chicken-or-egg problem. Our method is suited for images cropped along the head boundary, which is a much easier requirement on any system. Obtaining this cropped image requires a foreground/background separation technique. We have used GrabCut[14], available as an OpenCV function, on the output of a head tracker. For tracking, we have implemented a variation of the elliptical head tracker [15], which uses image gradient and color histogram for tracking a head through different poses in a video. A typical result from our tracker is shown in Fig. 9. Definite foreground and background regions are automatically marked for GrabCut, and we get cropped images as shown. Once the image is cropped along the head boundary, the same NPR technique used during training is applied, with the same threshold setting. The probability $P_(ij)$ that any pixel $i$ of value $x_i$ belongs to a specific pose $j$ can be found using equation (3).

$$P_{ij} = \frac{1}{\sigma_{ij}\sqrt{\pi}} e^{-\frac{(x_{ij} - \mu_{ij})^2}{2\sigma_{ij}^2}} \tag{3}$$

$$P_j* = max_j \prod_i P_{ij} \tag{4}$$

The estimated pose $j*$ maximizes the product of probabilities across all pixels according to equation (4). The same result can be achieved by finding the minimum log likelihood sum and thus saving on computation, as shown in equation (5).

$$L_{j*} = min_j \Sigma_i \frac{(x_{ij} - \mu_{ij})^2}{2\sigma_{ij}^2} \tag{5}$$

## 6 Results and Discussions

We feel that coarse pose is best expressed as lying in a window, rather than one angle with an unknown associated uncertainty. Thus, the results of the maximum likelihood estimation can be further interpreted as follows:

(i) First, the log likelihood sums for frontal, pan 30 °, 60 °, and 90° are considered, and the least among these fixes one end of the output pose range.
(ii) Next, the log likelihood sums for pan 15 °, 45 °, and 75 ° are considered, and the least among these fixes the other end.

For instance, if pan right 30 ° had the least log likelihood sum in step (i), and pan right 15° had the least log likelihood sum in step (ii), the pose is classified as between 15° to 30° turned right, i.e., 22.5° ± 7.5° pan right. Given the pan window, we compare the tilt-up and tilt-down log likelihood sums at that pan window to estimate tilt angle. For the window of 0° to 30°, tilt models are built at pan 15°; for the window of 30° to 60°, at pan 45°; and for the window of 60° to 90 at pan 75°.


**Fig. 9.** Using the tracker (left); real life results after using the tracker(right)

### 6.1 Performance Evaluation

The results we obtain for five-fold cross validation on the Pointing Database are encouraging, as shown in Tables 1 and 2. We show results for both methods i.e. regional segmentation and stroke rendering. The results show that regional segmentation works much better at pan angles less than 60°, implying that rough regional information corresponding to hair and facial features (e.g., eyes, and mouth) are the more appropriate features for head pose estimation in this range.However, at extreme pan angles, stroke rendering seems to perform better. This seems strange but actually at extreme side poses, there are not many regions which can be segmented out. The edges(eg jaw line) however become more prominent in the profile. So, stroke rendering performs better. [15] suggests that colour and gradient data are complimentary to one another. So we tried a third method wherein we multiplied the probabilities obtained using Regional Segmentation and Stroke Rendering and then rank ordered the now obtained probabilities in the same way as described above. As expected, these results are consistently accurate throughout the span of angles.

### 6.2 Comparisons with Other Approaches

To prove that an appropriate form of image abstraction is essential for a problem, we present cross-validation results without any form of abstraction i.e. direct down sampling of the image to size 32X32. Comparing with these results, the better performance of the abstraction methods is evident. Wu and Toyama (2000)[12] have also used Naive Bayes for estimating coarse head pose. Their method uses features extracted by three Gabor filters and a Gaussian filter, which is computationally much more expensive than our regional segmentation method, whose complexity is only of the order of the number of image pixels. We present cross validation results for our implementation of their system in Table 1. As we can see out system outperforms them in almost all cases. We also obtained Mean Average Errors and compared them with those of 3 other approaches. Since other approaches express it as the error w.r.t ($\pm$ 0°), and ours is w.r.t ($\pm$ 7.5°), we added half of 7.5 i.e. 3.75 to our errors(assuming a mean average error of 3.5 each time). This is shown in Table 3. Note that our approach performs consistently across all pan angles upto 90° and all tilt angles upto 60°

**Table 1.** Results for Pan(0°-90°) with tilt between 0°-30°

| | 0°-15° | 15°-30° | 30°- 45° | 45°-60° | 60°-75° | 75°-90° | Avg |
|---|---|---|---|---|---|---|---|
| Colour Segmentation | 92 | 92 | 96 | 88 | 82 | 80 | 92 |
| Stroke Rendering | 76 | 72 | 84 | 60 | 92 | 88 | 79 |
| Toyama et al[16] | 90 | 85 | 71 | 52 | 51 | 45 | 70 |
| Hybrid | 87 | 85 | 88 | 80 | 86 | 84 | 87 |
| Unabstracted Images | 85 | 80 | 80 | 71 | 70 | 65 | 77 |

**Table 2.** Results for Tilt(0°-60°)given pan angle range

| Pan Angle Range | 0°-15° | 15°-30° | 30°-45° | 45°-60° | 60°-75° | 75°-90° | Avg |
|---|---|---|---|---|---|---|---|
| Colour Segmentation | 80 | 84 | 87 | 84 | 82 | 80 | 84 |
| Stroke Rendering | 72 | 70 | 74 | 76 | 78 | 82 | 75 |
| Hybrid | 78 | 80 | 82 | 82 | 80 | 80 | 82 |

**Table 3.** Mean Average Error°

| Method | Pan | Tilt |
|---|---|---|
| Colour Segmentation | 4.95 | 6.15 |
| Stroke Rendering | 6.85 | 7.5 |
| Hybrid | 5.65 | 6.55 |
| Tu et al[17] | 14.1 | 14.9 |
| Gourier et al [18] | 10.1 | 15.9 |
| Dahmane et al[19] | 5.7 | 5.3 |

## 7 Conclusion and Future Work

We have discussed how the human brain uses much less information than what is actually available to it for processing images. We have tried to replicate the same and, in specific, demonstrated how NPR based image abstraction can be very useful for rough head pose estimation. Our algorithm is robust to illumination, expressions and identity and works successfully even at low resolutions, making it suitable for surveillance based applications. Moreover, it has little training effort or constraints and works in real time, enabling use on mobile platforms. This can also benefit fine pose estimation algorithms by providing them with an initial rough pose window. For instance, certain ML-based fine pose methods cannot be used practically because they need good initialization eg.bunch graphs [5].We believe that a technique with better color segmentation or other suitable forms of stylization and more sophisticated classifiers will further enhance these results and applications.

## References

1. Murphy-Chutorian, E., Trivedi, M.: Head pose estimation in computer vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 607–626 (2009)
2. Cai, Q., Sankaranarayanan, A., Zhang, Q., Zhang, Z., Liu, Z.: Real time head pose tracking from multiple cameras with a generic model. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 25–32 (2010)
3. Saragih, J., Lucey, S., Cohn, J.: Face alignment through subspace constrained mean-shifts. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1034–1041 (2009)
4. Huang, D., Storer, M., De la Torre, F., Bischof, H.: Supervised local subspace learning for continuous head pose estimation. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921–2928 (2011)
5. Fu, Y., Huang, T.: Graph embedded analysis for head pose estimation. In: 7th International Conference on Automatic Face and Gesture Recognition, FGR 2006, pp. 3–8 (2006)
6. Cole, F., Sanik, K., DeCarlo, D., Finkelstein, A., Funkhouser, T., Rusinkiewicz, S., Singh, M.: How well do line drawings depict shape? In: ACM SIGGRAPH 2009 papers, SIGGRAPH 2009, pp. 28:1–28:9. ACM, New York (2009)
7. Ryan, T.A., Schwartz, C.B.: Speed of perception as a function of mode of representation. American Journal of Psychology 69, 60–69 (1956)
8. Cavanagh, P.: Vision is getting easier everyday. Perception 24, 1227–1232 (1995)
9. Winnemöller, H., Feng, D., Gooch, B., Suzuki, S.: Using npr to evaluate perceptual shape cues in dynamic environments. In: Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering, NPAR 2007, pp. 85–92. ACM, New York (2007)
10. Zeki, S.: A vision of the brain. Blackwell Scientific Publications, Oxford (1993)
11. Strothotte, T., Schlechtweg, S.: Non-photorealistic computer graphics: Modeling, rendering, and animation. Morgan Kaufmann (2002)

12. DeCarlo, D., Santella, A.: Stylization and abstraction of photographs. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2002, pp. 769–776. ACM, New York (2002)
13. Gourier, N., Hall, D., Crowley, J.L.: Estimating face orientation from robust detection of salient facial features. In: Proceedings of Pointing, ICPR, International Workshop on Visual Observation of Deictic Gestures, vol. 1, pp. 617–622 (2004)
14. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23, 309–314 (2004)
15. Birchfield, S.: Elliptical head tracking using intensity gradients and color histograms. In: Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 232–237 (1998)
16. Wu, Y., Toyama, K.: Wide-range, person- and illumination-insensitive head orientation estimation. In: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 183–188 (2000)
17. Tu, J., Fu, Y., Hu, Y., Huang, T.: Evaluation of Head Pose Estimation for Studio Data. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 281–290. Springer, Heidelberg (2007)
18. Gourier, N., Maisonnasse, J., Hall, D., Crowley, J.L.: Head Pose Estimation on Low Resolution Images. In: Stiefelhagen, R., Garofolo, J.S. (eds.) CLEAR 2006. LNCS, vol. 4122, pp. 270–280. Springer, Heidelberg (2007)
19. Dahmane, M., Meunier, J.: Object representation based on gabor wave vector binning: An application to human head pose detection. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 2198–2204 (2011)