Robust color-based skin detection for an interactive robot

Alvise Lastra, Alberto Pretto, Stefano Tonello and Emanuele Menegatti

Department of Information Engineering
via Gradenigo 6/b, 35131 Padova, Italy
lastraal@dei.unipd.it

Abstract. Detection of human skin in an arbitrary image is generally hard. Most color-based skin detection algorithms are based on a static color model of the skin. However, a static model cannot cope with the huge variability of scenes, illuminants and skin types. This is not suitable for an interacting robot that has to find people in different rooms with its camera and without any a priori knowledge about the environment nor of the lighting.
In this paper we present a new color-based algorithm called VR filter. The core of the algorithm is based on a statistical model of the colors of the pixels that generates a dynamic boundary for the skin pixels in the color space. The motivation beyond the development of the algorithm was to be able to correctly classify skin pixels in low definition images with moving objects, as the images grabbed by the omnidirectional camera mounted on the robot. However, our algorithm was designed to correctly recognizes skin pixels with any type of camera and without exploiting any information on the camera.
In the paper we present the advantages and the limitations of our algorithm and we compare its performances with the principal existing skin detection algorithms on standard perspective images.

1 Introduction

The identification of people represented into images or videos is a challenging problem addressed since many years. The applications of a reliable and robust algorithm for people detection in any kind of images can be virtually unlimited. Techniques and theoretical assertions were presented, but most of them give reliable results only with structured settings or with "a priori" fixed imaging conditions. Moreover, the most reliable solutions require specific and expensive hardware-software resources. The aim of this work is a general technique that correctly recognizes skin pixels independently on the different ethnic groups, under varying illumination conditions in whatever complex environment, only using chromatic informations. The result is the development of a new complex, but fast end efficient to compute filter, we called it VR Filter.

This work was motivated by the creation of a robust and reliable skin detection algorithm to be used as main input for the "people finding module" of the software

architecture controlling the robot in Fig. 1. This work is the result of the meeting of Robotics and Art. This is an interactive robotic sculpture conceived and realized by the artist Albano Guatti. The robotic part was totally developed by people at the IAS-lab and at IT+Robotics according to Guatti's concept. The robot's main sensor is an omnidirectional camera (well integrated with the artistic appearance of the statue). The omnidirectional camera is used to detect the persons in the environment, thanks to the skin detection algorithm described in this paper. The omnidirectional visual perception is coupled to an omnidirectional range sensor realized with a ring of Polaroid sonar sensors.

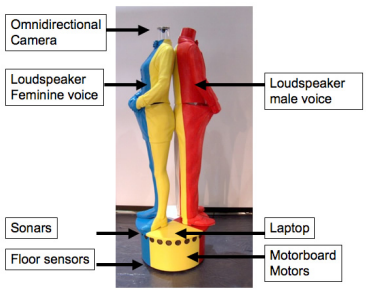


Fig. 1. The interactive robotic sculpture by Albano Guatti

2 The Skin detection problem

2.1 Definition of problem

First of all we shall formalize the skin detection problem as generally as possible Let be `P` the following problem we are going to solve: `P`:
Given `I(R,G,B)`, in the following simply `I`, an arbitrary image we don't know anything about it (which are its contents, type of source and the environment conditions when it has been generated), we want to identify all the regions and only the regions Ω of `I` where human skin is present.
In particular, we want to be able to successfully process low definition images with moving objects in very complex scenarios as usually are the omnidirectional images grabbed by mobile robots.

2.2 Related work

As mentioned in the introduction, the skin detection problem is still a very investigated problem; many authors have proposed techniques to solve it by fixing one or more parameters of the problem, but a solution of `P` considering all of them has

never been given. Soriano et al. [9] showed a camera-specific color-based method ables to recognize skin in different light conditions and proposed a database of camera behaviors to complete it. The use of a normalized color space, in this case the `rg` normalized color space, is interesting because it allow to isolate skin locus with simple quadratic functions. Also for [6], [7], [11], [12] a normalized color space, the `rg` normalized color space again (in the following simply `rg`), is the most effective to extract with success a skin locus. This is because it is as little as possible dependent on the illuminant. In addition, Albiol et al. [2] affirmed that an optimum filter for skin detection will have the same performance even working in different color spaces. Other authors suggested to solve the `P` problem proposing a union of different techniques to improve the results of a single color-based method and its defects; Kruppa et al. [4] and Tomaz et al. [11] used, for example, a color-based filtering with a shape identification obtaining good result for face detection. In [11] again and also in [3] a static prefilter on `RGB` space is used too: with this last kind of filters it is easier and more natural to remove zones that surely are non-skin areas (pixels too inclined to black, to green or to blue etc). At last we mention Lee et al. [5] who proposed an elliptical boundary for skin locus using a gaussian model and six chromatic spaces and Sebe et al. [8] who proposed a Bayesian network approach instead.

2.3 The VR Filter

As stated, the `P` problem is too wide, we need to insert some limitations. We need to introduce two constraints (that anyway do not compromise the generality of method itself):

$\mathbf{V}_1$ *The image* `I` *has to represent a scene not too obscure nor too saturated*
$\mathbf{V}_2$ *The source of* `I` *has to ensure that its calibration is not strongly unbalanced*

$\mathtt{V}_1$ excludes from `P` all images captured in illumination conditions near to darkness or saturation, while $\mathtt{V}_2$ excludes from `P` all images that have chromatic features too altered (e.g., images with a very high contrast).

(a) (b) (c) (d)

Fig. 2. Example images: (a) satisfies $\mathtt{V}_1$, (b) does not satisfies $\mathtt{V}_1$, (c) satisfies $\mathtt{V}_2$, (d) does not satisfies $\mathtt{V}_2$

From now on, `P` will be the initial problem, restricted by $\mathtt{V}_1$ and $\mathtt{V}_2$, and `I` will be an image satisfying $\mathtt{V}_1$ and $\mathtt{V}_2$.

In brief, the strategy of our method is the sequence of two distinct color-based techniques and could be called "catch and clean the skin locus". The first filter "catches" the skin locus, even capturing spurious pixels, while the second "cleans" possible false positive pixels selected by the first one. We chose this approach, because we experimentally obtained a dynamic region, depending on the statistics of first and second order of the image, ables to intercept the skin locus; the formal and mathematical expression of this region is the core of our work. So, our filter, called VR filter, is a cascade of two filters that we called V filter and Rm filter, respectively.

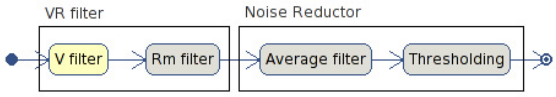


Fig. 3. UML flow of the algorithm implementing the VR filter

V Filter The V filter is a dynamic filter based on the definition of a 2D region of a color space that we called V region (Ω_V). Ω_V depends on the statistics of first and second order of `I`: let be `xy` a generic two-dimensional color space and let be f_x and f_y the distributions of `I` with respect to `x` and `y`, respectively. f_x and f_y can be considered as mass distributions of two discrete aleatory variables x and y. Thus, we can compute the expectation m (1) and the positive radix of the second order central moment σ (2):

$$m_x = \sum_{\alpha \in A_x} \alpha f_x(\alpha), \quad m_y = \sum_{\alpha \in A_y} \alpha f_y(\alpha) \tag{1}$$

$$\sigma_x = \left[\sum_{\alpha \in A_x} (\alpha - m_x)^2 f_x(\alpha)\right]^{\frac{1}{2}}, \sigma_y = \left[\sum_{\alpha \in A_y} (\alpha - m_y)^2 f_y(\alpha)\right]^{\frac{1}{2}} \tag{2}$$

where in (2) A_x and A_y are the alphabets of the two aleatory variables x, y

Let's now define a set, we called V_{bone} (γ_V), that will be helpful to understand the meaning of Ω_V:

$$\begin{gathered} \gamma_V = \left\{(x,y) :\ x < m_x,\ y = m_y,\ x \in \mathtt{x},\ y \in \mathtt{y}\right\} \\ \bigcup \\ \left\{(x,y) : y = \tfrac{\sigma_y}{\sigma_x}(x - m_x) + m_y,\ y < m_y,\ x \in \mathtt{x},\ y \in \mathtt{y}\right\} \end{gathered} \tag{3}$$

γ_V is the union of two half-rays of $\mathbb{R}^2$ with origin in (m_x, m_y) the first with angular factor equals to zero, the second with a non-negative one. Finally we define Ω_V as the union of two half-stripes described by the following formulas:

$$\Omega_V = \big\{(x,y):\ x < m_x,\ |y - m_y| < \sigma_y,\ x \in \mathtt{x},\ y \in \mathtt{y}\big\}$$

$$\bigcup \tag{4}$$

$$\left\{(x,y) : \left|y - \tfrac{\sigma_y}{\sigma_x}(x - m_x) - m_y\right| < \sigma_y,\ y < m_y + \sigma_y,\ x \in \mathtt{x},\ y \in \mathtt{y}\right\}$$

Intuitively, Ω_V appears, in the generic `xy` plane, as a "V" rotated counter-clockwise of about $\pi/2$. In Fig. 4, we plot three examples of γ_V (blue lines) with corresponding Ω_V (red areas) in a generic `xy` normalized color space generated by three different images.

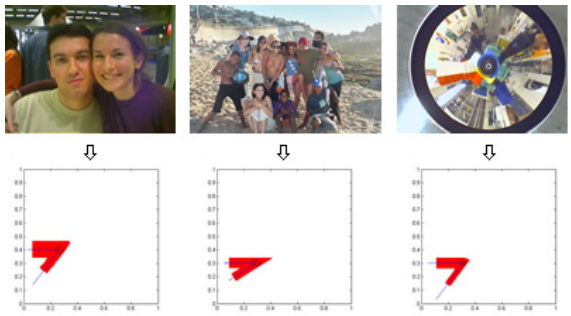


Fig. 4. Examples of different γ_V and Ω_V

Both γ_V and Ω_V can be create in any 2D-space, but their usefulness for our goal is that we have experimentally verified that *if the 2D-space* ***xy*** *is the bg normalized color space, the γ_V intercepts the skin locus for each* `I` *of* `P`*. Therefore, in the* `bg` *normalized color space, Ω_V contains at least a part of the skin locus for each* `I` *of* `P`*.*
Thus, V filter works in the `bg` normalized color space; we recall that the `bg` normalized color space is defined from the `RGB` color space as:

$$b = \frac{B}{R+G+B}, \quad g = \frac{G}{R+G+B} \tag{5}$$

so defined w and h as the width and the height of `I`, respectively, the V filter can be defined by:

$$V(i,j) = \begin{cases} 1 & if\ \mathtt{bg}(i,j) \in \Omega_V \\ 0 & otherwise \end{cases} \quad with\ \ 0 < i \leqslant w,\ 0 < j \leqslant h$$

Rm Filter The Rm filter is a static filter. It works in the `RGB` color space and has been designed to remove regions of the color space that the V filter might has

selected and that with high probability not belong to the skin locus.
Rm filter is simply defined as:

$$Rm(\mathtt{R},\mathtt{G},\mathtt{B}) = \begin{cases} 1 \;\; if\ \mathtt{G} < k_G; \text{ AND } \mathtt{G} < \mathtt{R} - k_{RG} \text{ AND} \\ \quad m_R\mathtt{R} < \mathtt{G} < M_R\mathtt{R} \text{ AND } \mathtt{B} < \frac{\mathtt{R}+\mathtt{G}}{k_B} \\ 0 \;\; otherwise \end{cases} \tag{6}$$

The equations in 6 have been designed to remove, in this order, color tones that have too much green, too much green with respect to red and too much blue with respect to green and red.

To set the five constant parameters in (6) a directed reinforcement learning technique, called Counter-based with decay [10], has been used. As learning patterns, we have considered 25 images with different subjects and lighting conditions. The target of this technique was to maximize the test score we will define in (10).

So the optimal values for the parameters have been resulted the following:
$k_G = 166$; $k_{RG} = 25$; $m_R = 0.563$; $M_R = 0.808$; $k_B = 2.0$;

Finally we can define our VR filter as this mask:

$$VR = V \text{ AND } Rm \tag{7}$$

After *VR* (7) an average filter and a thresholding operation are applied to the output of *VR*, to stabilize the results and to remove noise around the selected regions; so, they appear more regular and are easier to process by any subsequent image processing algorithm (see Fig. 3).

3 Tests and Results

The tests are been executed on a dataset of over 500 images of different generic sources (pictures taken form omnidirectional cameras, the Internet, perspective cameras and videoframes) but also from all the "Georgia Tech Image Database" [1]. To better catalog all the images, they have been divided into seven categories:

Cat A: Subject in foreground with simple background
Cat B: Subjects in foreground with complex background
Cat C: Night indoor/outdoor environments with artificial lights
Cat D: Daily outdoor environment with difficult scene or lighting conditions
Cat E: Different ethnic groups
Cat F: Complex omnidirectional images
Cat G: Complex omnidirectional images with moving subjects

3.1 Test metrics definition

To test and to measure the perform of our filter we have design some formal rules.

Let's consider two B&W images, the first generated by the VR Filter as the mask of the filtered output and the second that represents the mask of the skin pixels manually extracted from the original image. Let be M_{VR} and M_{HR} respectively. Both these images are in binary encoding: for M_{HR}, as example,

$M_{HR}(x, y) = 1$ if the pixel (x,y) is considered a skin pixel, $M_{HR}(x, y) = 0$ otherwise. From M_{VR} and M_{HR} is computed a new image T:

$$T = M_{HR} - M_{VR} \tag{8}$$

Each pixel of T can assume only three values:

-1 if the pixel is a non-skin pixel recognized as a skin pixel (false positive [FP])
0 if the pixel, either skin or non-skin, is correctly recognized (recognized [OK])
1 if the pixel is a skin pixel not recognized (miss [MS])

From T are successively computed three parameters:

$k_{OK} = \#(0)$ in T; $k_{FP} = \#(\text{-}1)$ in T; $k_{MS} = \#(1)$ in T

Finally, defined $N = w \cdot h$ where w and h are the same defined in 2.3 and $k_{M_{HR}} = \#(1)$ in M_{HR}, we compute the following result test values:

$$p_{OK} = \frac{k_{OK}}{N}, p_{MS} = \frac{k_{MS}}{k_{M_{HR}}}, p_{FP} = \frac{k_{FP}}{N - k_{M_{HR}}} \tag{9}$$

and a resume test score as:

$$S = 2p_{OK} - 5p_{MS} - p_{FP} \tag{10}$$

With the values defined in (9) a strict test conclusion can be given as reported in Table I, so a test results a positive match, if and only if, at least the 75% of pixels are correctly recognized, and the each of the percentages of the skin pixel and of the non-skin pixels that have been correctly recognized is over 80 % and 90 % respectively.

$p_{OK} \geqslant 0.75$	$p_{FP} < 0.10$	$p_{FP} \geqslant 0.10$
$p_{MS} < 0.20$	Correct match	Correct match with too false positives
$p_{MS} \geqslant 0.20$	Miss	Miss

Table 1. Test result based on values of (9) with $p_{OK} \geqslant 0.75\%$. All tests having $p_{OK} < 0.75\%$. are labeled as Miss

In Figure 5 we show an example of visual test result.

3.2 Algorithm performance and statistical results

All operations executed by the VR filter are linear in the image dimensions; thus, its computational complexity is $\Theta(w \cdot h)$.

In Table II we reported the statistical test results by apply the VR filter on the dataset of images, while Table III shows the processing time spent by our algorithm in a C/C++ implementation.

Fig. 5. From left to right and from top to bottom: original image (s13/13.jpg of [1]), manually extracted skin mask, VR filter output and graphical output of the test. In this last image the skin and the non-skin pixels correctly recognized are respectively green and lime, the FP pixels are blue and the MS pixels are red. For this image we have: OK = 92.62%; MS = 14.67% and FP = 2.87%; S = 1.56

Cat	Positive match	Positive match with too false positives	Miss cases
A	87.23 %	9.04 %	3.73 %
B	88.00 %	10.00 %	2.00 %
C	86.00 %	8.00 %	6.00 %
D	82.00 %	12.00 %	6.00 %
E	85.18 %	9.08 %	5.74 %
F	90.00 %	8.00 %	2.00 %
G	88.00 %	8.00 %	4.00 %
Total	**86.63 %**	**9.16 %**	**4.21 %**

Table 2. Summary of test result's percentage by category

The percentage of hit is very high on images with normal lighting conditions, even if there are complex scenes, and is lower, but still good, on night images. Using a resolution of 800x600 is possible to compute up to 2.5 frame per second; this rate is not very high, however is higher then most alternative techniques proposed in the literature. To speed-up the computation of a sequence of video frames, the VR filter can be used to create a look-up table (LUT) containing the 3D region of the `RGB` color space that contains the skin locus of the first frame; the subsequent frames can be processed accessing the LUT to check if the pixels belong to the skin locus or not. The LUT needs to be updated by VR filter only if the lighting conditions change in time.

3.3 Some tests on generic images

In this section we present the results of our skin detector on some images grabbed with the omnidirectional camera of our robot (Fig. 6), generic images grabbed with

Image	C/C++		Image	C/C++	
resolution	kpxlps	fps	resolution	kpxlps	fps
320x240	1182	15.36	1024x76	1311	1.67
640x480	1258	4.07	1280x1024	1327	1.22
800x600	1280	2.67	1600x1200	1319	0.69

Table 3. Computation time - test on Pentium M 1.5GHz

a digital camera (left of Fig. 8) and obtained from the Internet (other images). Figures have been organized into categories as explained in Section 3.

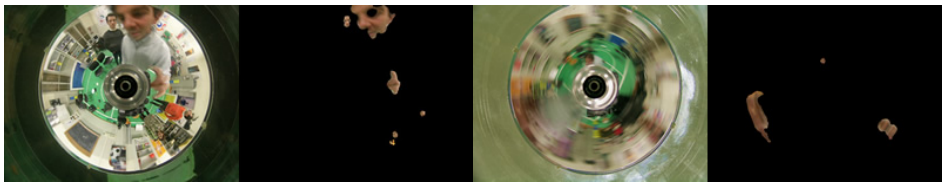


Fig. 6. On the left, an example of Cat. F: positive match 90.00 %. Picture grabbed by the interactive robotic sculpture of Figure 1 and an example of Cat. G: positive match 88.00 %, on the right

Fig. 7. On the left, an example of Cat. A: positive match 87.23 % (this example refers to the image s03/04.jpg of [1]) and, on the right, two examples of Cat. B: positive match 88.00 %

Finally we report a comparison between our VR filter with some skin detector proposed by other authors (Fig. 10–12). We used the original images extracted by the cited papers.

4 Conclusions and Future works

A new color-based skin detection algorithm has been presented. Our approach gives a solution for the skin detection problem, in conditions as generic as possible

Fig. 8. Examples of Cat. D: positive match 82.00 %

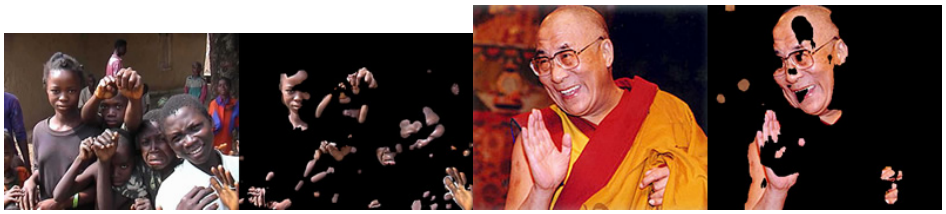


Fig. 9. Examples of Cat. E: positive match 85.18 %

and it uses only chromatic information as input. As reported in the literature, the use of one color space is not enough for arbitrary images and a combined solution is needed. The result of our work is the VR filter; it is composed of a cascade of two filters: the V filter and Rm filter. The first is a dynamic filter working in the `bg` normalized color space. The latter is a static filter working in the `RGB` color space. This technique is robust and reliable, if the input image satisfies two constraints $\mathtt{V}_1$ and $\mathtt{V}_2$ (that anyway do not compromise the generality of method itself).

We compared the performance of the VR filter with various skin detector (color-based and not) and our method gave comparable or better results, even if it uses a simpler and faster technique; it also works correctly with a larger range of images.

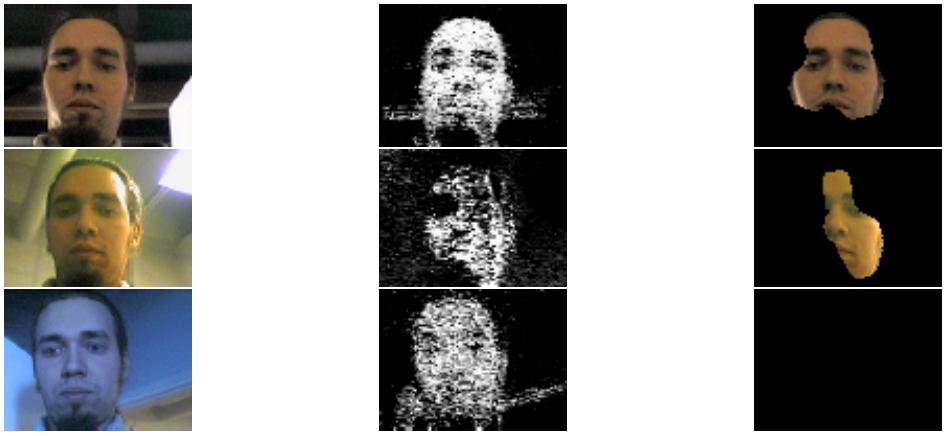
The proposed VR filter has been successfully used in several exhibitions of the interactive robotic sculpture of Fig. 1. The robot run for five days at SMAU 2005 (the biggest Information Technology fair in Italy) moving around among hundreds of persons. At MART (Museum of Modern Art, Rovereto (TN) Italy the robot run for two days in the cafeteria and in the museum hall.

Future works will be aimed at relaxing the assumption $\mathtt{V}_1$ and $\mathtt{V}_2$, in order to be able to correctly process any images. For this scope we are working to remove the static numerical parameters of the filter, by making the Rm filter dynamic.

References

1. *Georgia Tech Face Database.*
   ftp://ftp.ee.gatech.edu/pub/users/hayes/facedb/.

Comparison 1 Soriano et al. technique [9] vs VR filter:



(a) (b) (c)

Fig. 10. Original images (a), Soriano filter (b) and VR filter (c). Soriano's camera-specific technique is able to correctly recognize skin pixels under incandescent and fluorescent lamps, while VR filter is camera independent and less sensitive to noise, but misses the face under fluorescent light.

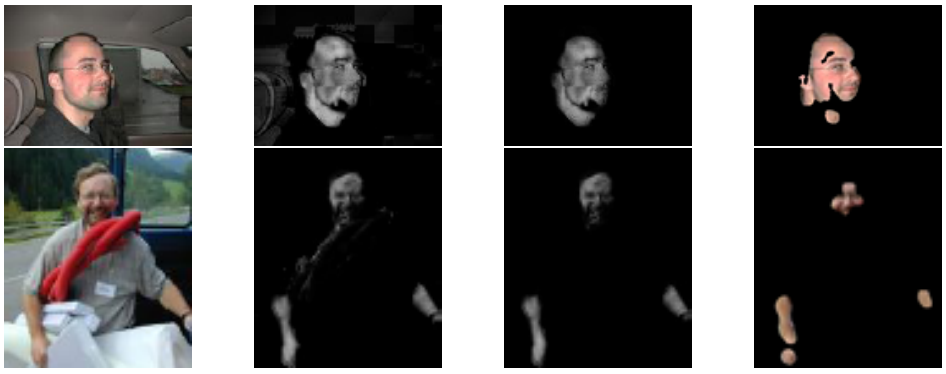
2. A. Albiol, L. Torres, and E. J. Delp. Optimum color spaces for skin detection. In *International Conference on Image Processing*, volume 1, pages 122–124, 2001.
3. R. Kjeldsen and J. Kender. Finding skin in color images. In *2nd International Conference on Automatic Face and Gesture Recognition*, pages 312–317, 1996.
4. H. Kruppa, M. A. Bauer, and B. Schiele. Skin patch detection in real-world images. *Annual Symposium for Pattern Recognition of the DAGM 2002*, pages 109–117, 2002.
5. Y. J. Lee and I. S. Yoo. An elliptical boundary model for skin color detection. In *International Conference on Imaging Science, Systems, and Technology*, pages 472–479.
6. B. Martinkauppi. *Face colour under varying illumination - analysis and applications*. PhD thesis, University of Oulu, 2002.
7. K. Schwerdt and J. Crowley. Robust face tracking using color, 2000.
8. N. Sebe, I. Cohen, T. S. Huang, and T. Gevers. Skin detection: A bayesian network approach.
9. M. Soriano, S. Huovinen, B. Martinkauppi, and M. Laaksonen. Skin detection in video under changing illumination conditions. In *15th International Conference on Pattern Recognition*, volume 1, pages 839–842, 2000.
10. S. B. Thrun. Efficient exploration in reinforcement learning. Technical Report CMU-CS-92-102, Pittsburgh, Pennsylvania, 1992.
11. F. Tomaz, T. Candeias, and H. Shahbazkia. Improved automatic skin detection in color images. In *7th Digital Image Computing: Techniques and Applications*, pages 419–427, 2003.
12. T. Wilhelm, H. J. Bhme, and H. M. Gross. A multi-modal system for tracking and analyzing faces on a mobile robot. In *Robotics and Autonomous Systems 48*, pages 31–40, 2004.

Comparison 2 Tomaz et al. technique [11] vs VR filter:

(a) (b) (c)

Fig. 11. Original images (a), Tomaz filter (b) and VR filter (c). VR filter is more robust to highlights (first row) and to background noise (second row), in addition Tomaz el al. method also needs an initial camera calibration.

Comparison 3 Kruppa et al. technique [4] vs VR filter:



(a) (b) (c) (d)

Fig. 12. Original images (a), Kruppa color-based filter (b), Kruppa color+shape filter (c) and VR filter (d). VR filter and Kruppa's color+shape algorithm results are similar; comparing the performance of the two algorithms, VR performs better.