

A System for Automatic Coin Identification and Classification

Ștefan Nicolae Tică, Costin-Anton Boianțiu, Ion Bucur

University “Politehnica” of Bucharest, Splaiul Independenței 313, Bucharest, 060042, Romania

stefan.tica@cti.pub.ro,
costin.boianțiu@cs.pub.ro,
ion.bucur@cs.pub.ro

Abstract. In this paper we would like to present and discuss a system that automatically classifies coins. This flexible system can identify coins having different features and being photographed in different light conditions. For this purpose a set of strong techniques for thresholding, edge detection and frequency domain processing were used in order to generate a fingerprint as meaningful and as invariant as possible for every coin class. Usually, the capturing of digital images cannot be performed in best conditions and inconsistencies can arise due to various lighting conditions as well as the performance of the capturing device. This article is here to suggest a method to reduce problems generated by lighting, so that image characteristics are more accurate. The proposed solution improves an existing automatic coin classification algorithm by applying illumination correction before the actual classification.

Keywords: Canny edge detection, coin classification, DCT transform, lighting invariant, Otsu thresholding, rotation invariant.

1 Introduction

There are institutions working with historical coins, which could use a system for automatic coin classification. A system like this can also be useful to historians or even private coin collectors or other people interested in this domain. A free online service, able to perform clustering on coins based on their similarities is very useful for the community. The same system can also be used in sorting or classification of the various types of coins that were collected in the European countries after the euro was released.

2 Related Work

In the modern scientific literature are present various approaches regarding the subject of coin recognition.

A system based on a rotation-invariant neural network was described in [8] as a system capable of identifying some Japanese coins (a 500 yen and a 500 won piece). By generating the rotational group for a coarse model of the coin and collecting all the facts into a neural network, can be achieved the rotational invariance. The neural network method has some drawbacks, for example, the rules for coin rejection are not very clear. It is very important to be able to reject coins, but in the same time, there is no possibility of knowing in advance which kind of coins will be given to the system.

Strategies like decision trees [15], neural network and Bayesian classifiers were compared in [15] and [4]. They obtained a form of the decision tree algorithm [6] capable of rejecting coins in case that their defining attributes were outside an acceptance range. As a drawback, this method is difficult to extend to images.

In [1] is presented an approach based on color images. Translational invariance was obtained through segmentation and rotational invariance from using polar coordinate representation and correlation.

3 Feature extraction

Using more characteristics extracted from coin images, the precision in the identification of similar coins can be increased. [18] introduces three main features related to coin images:

- the picture on the coin (stamp)
- the texture on the coin
- the text on the coin

This approach concentrates on the first two characteristics. For a proper measurement of the subsequent features, a coin image preprocessing phase is needed, so that the inserted images may reflect as precise as possible uniform lighting conditions and contrast normalization. One such very important preprocessing step in coin analysis is the removal of the specular reflection, which strongly alters almost all the coin photos.

3.1 Specular Reflection Removal

Usual specular removal methods work only on color images (the analysis of chromaticity, for example) in order to deduce the intensity of the specular component and for diminishing it. [22] Referring to the fact that all coins are made from a single material, we can affirm that they are in most cases monochrome, so, usual methods do not work, and thus by processing them in grayscale the quality of classification will not change.

The specular component can be obtained in luminance space, by the binarization of the image (fig.1). Therefore, the final image that is going to be processed will result by subtracting that specular component from the L component.

This step has to be performed before the actual edge detection, in consideration of finding a more uniform edge distribution.



Fig. 1. Original image (left) compared with the preprocessed (right)

3.2 Stamp (picture) Representation

The stamp is the most important part of a coin's fingerprint. This is based on the statistical distribution of the edges that appear on the surface of the coin. The extraction of the edge characteristic is a process consisting three steps:

- edge detection
- edge distance distributions
- edge angle distribution

In the papers [11,12] it is presented a similar method describing the edge distance and angle distributions.

Edge detection.

Based on our practical tryouts and on presentation from [16] of a number of various edge detection algorithms, we came to the conclusion that the most indicate one for this particular project is Canny's edge detection algorithm. We also carried experiments with the Sobel kernel edge detection, Prewitt's operator, Robert's cross operator. Other algorithms exist, which would be able to improve this edge detection [20].

The algorithm of Canny has six steps [9]:

Step 1.

Filter out any noise in the original image with the purpose to avoid detection of false edges. This is done by applying a Gaussian filter or other filters for specific purposes [5]. The Gaussian used in the implementation is presented in Fig. 2.

$$\frac{1}{115}$$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Fig. 2. Discrete approximation to Gaussian function with $\sigma=1.4$

Step 2.

Compute the gradient of the image in order to determine the edge strength by first applying the Sobel operator on the image to compute a 2-D spatial gradient on both axes and then approximating the absolute gradient magnitude (edge strength) for each point. The magnitude is approximated using the next formula:

$$|G| = |G_x| + |G_y| \quad (1)$$

where G_x and G_y stand for axis oriented gradients. Those were gained by applying two 3x3 convolution masks (like in Fig. 3):

-1	0	+1
-2	0	+2
-1	0	+1

G_x

+1	+2	+1
0	0	0
-1	-2	-1

G_y

Fig. 3. G_x and G_y

Step 3.

Computing the direction of the edge using the gradient in the X and Y directions:

$$\theta = \arctan(G_y / G_x) \quad (2)$$

Step 4.

Now the edge direction is computed, so the next thing to do is to relate the edge to a direction that can be traced in an image. Given a 5x5 image as the one described in Fig. 4 and analyzing pixel "P" it can be seen that there are only four possible directions when describing the surrounding pixels - 0 degrees (for the horizontal direction), 45 degrees (along the main diagonal), 90 degrees (for the vertical direction), or 135 degrees (along the secondary diagonal).

```

x  x  x  x  x
x  x  x  x  x
x  x  P  x  x
x  x  x  x  x
x  x  x  x  x

```

Fig. 4. Finding possible directions

Step 5.

With the edge direction estimated, non-maximum suppression is applied along the edge and removes any border pixel, so the result is a thin line in the output image.

Step 6.

At the end, a hysteresis processing is employed as a means of eliminating potential streaking. This streaking, which is the breaking of an edge contour, might occur due to the operator output fluctuating above and below the threshold. Hysteresis doesn't use a single threshold, but two: a high one and a low one. A pixel with a value greater than high threshold T1 is presumed to be an edge pixel and is marked as such instantly. A pixel that is connected to this edge pixel with a value greater than the lower threshold T2 is also selected as an edge pixel. Following an edge can therefore be expressed as finding a gradient of at least T1 and continuing as long as the value does not fall below T2. The method can be improved as indicated by [7].

These two thresholds can be calculated using Otsu's algorithm, an approach that represents a binarization algorithm [14], widely used, and which has spawned many forms and adaptations [2], [10], because of its stability and high speed. It is a threshold based, global binarization algorithm, which tries to maximize the inter-class variance (or minimize the intra-class variance). The inter-class variance σ_b^2 is defined as follows:

$$\sigma_b^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (3)$$

where $\omega_x(t)$ represent the class probability, and $\mu_x(t)$ represents the class mean. Those are deducted by the following formulas:

$$\omega_x(t) = \sum_{i=0}^t c(i) \quad (4)$$

$$\mu_x(t) = \sum_{i=0}^t c(i) \cdot i \quad (5)$$

where $c(i)$ stands for the number of pixels in the image with the value i . So the equation tries to separate the means of the cluster, while also keeping each cluster with a high probability of occurrence.

The purpose of Otsu's method is to obtain the threshold t^* such that:

$$t^* = \arg \max_t \sigma_b^2(t). \quad (6)$$

An exhaustive search is employed. For many optimal thresholds the accepted value is the mean value [19]. This is very important for the quality of the binarization after the post-processing threshold stage.

The final step of Canny's edge detector uses two thresholds. The best values for them can be experimentally determined for a given image, but the method cannot be considered for our project because we need to elaborate a way to set these values automatically.

We can find a solution for this by determining the threshold with Otsu, for a given coin image, deriving our thresholds T1 and T2 based on this value. We experimentally deduced that T1 is equal to 50% of the threshold of Otsu and T2 equal to 30% of the same parameter.

Some problems occurred during the experiments regarding medieval and older coins classification: those types of coins mostly have an unregulated, bigger stamp height, a characteristic that is not met on modern coins which tend to be flattened. The light condition and the way shadows are formed when taking the picture, may create the possibility to obtain different pictures of the same coin with totally different results regarding the edges detected using Canny's edge detector.

The program must also overcome the diffuse illumination variation problem, because the presented work considers photographs taken in an uncontrolled environment. A method proposed by [3] is based on transforming the image from the RGB space to LUV space. A modern coin image and a medieval one transformed in LUV space and its three composing parts: The L, U and V image components were obtained by scaling the LUV values to 0-255 is shown in fig. 5.



Fig. 5. A modern (upper row) and medieval (lower row) coin picture a) original picture; b) L-component; c) U-component; d) V-component

Using the Canny's edge detector on the image L-component we gained better results than using the grayscale representation of the image (Fig. 6).

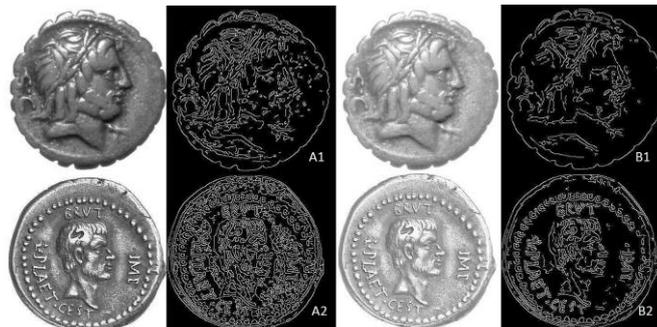


Fig. 6. Grayscale image and corresponding Canny's edges (A1, A2) Image's L-component and associated Canny's edges (B1, B2)

Edge Distance Distribution.

The distribution of the distances between the pixels on the edges and the center of the coin is measured by the edge distance distribution. This is obtained by partitioning the coins in circular concentric parts, as it is shown in fig.7. The number of edge pixels in each part is accumulated, and the resulting histogram is normalized for providing an estimation of the edge distance distribution. Edge distance distributions are rotation invariant by definition. Edge distance distributions are used in a multiscale method, by measuring the histograms for different numbers of bins (for 2, 4, 8, 16, and 32 bins).

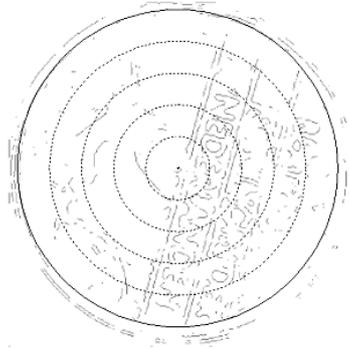


Fig. 7. Distance distribution

Edge Angle Distribution.

Not all the information are incorporated in edge distance distributions, although those instruments appear to be very powerful and occupying an important place in coin classification. For example, the relative angular distribution of the edge pixels is not represented. This can be expressed using the edge angle distributions, which are measured by dividing the coin in pie-shaped parts, like represented in fig. 8. The edge pixels presents in the part are accumulated and the histogram that results is normalized in order to provide an estimation of the relative angular distribution of the coin edge pixels.

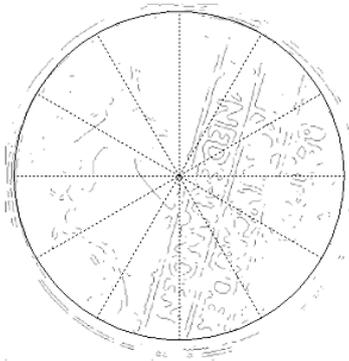


Fig. 8. Angle distribution

The difference between the edge distance distributions and edge angle distribution is that the second ones are not rotation invariant by definition. Rotation invariance of the edge angle feature can be obtained by computing the magnitude of the Fourier transform of the obtained histogram. This stage makes the histogram invariant under circular shifts (which correspond to rotations of the coin). In this respect, a large number of bins in the histogram are required, since a rotation of the coin should imply a circular shift on the histogram, instead of a change in the histogram accumulators.

In this application the edge angle histogram is measured in scales of 240 bins, compressing a spatial signal by transforming it to the frequency domain and dropping high-order values and keeping low-order ones.

Texture of the Coin.

The texture component of the fingerprint is based on two DCT (Discrete Cosine Transform) parts of the LUV representation of the coin, image inspired by the work of [13].

The discrete cosine transform makes possible the splitting of the image into sections/ spectral sub-bands of differing importance (respecting the image's visual quality). The DCT makes the transformation of a signal from the spatial domain into the frequency domain. The values obtained by applying the DCT have coincidentally order from the lowest to highest frequency. This characteristic and the psychological observation that the human eye is less sensitive to recognizing the higher-order frequencies, makes the lead to the possibility of compressing a spatial signal by transforming it to the frequency domain and dropping high-order values and keeping low-order ones.

A 2D-DCT is calculated over the luminosity (L-channel) of the image [17]. The first coefficient represents the DC value, or average luminosity of the image. The other one represent the higher order values with increasing frequency. Some of these coefficients (the first 10 of them) are normalized for the grayscale fingerprint section. This part of the fingerprints represents the basic composition of the image.

After that, a 2D-DCT of the two color components are calculated and used for the color part of the fingerprint. In this case, just the first three of each are taken into consideration, since the human eye is much more sensitive to luminosity than to color. This part of the fingerprint represents the color composition of the image, with reduced spatial resolution compared to the gray scale part.

The 2D DCT is given by the formula:

$$X_{k_1, k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[\frac{\pi}{N_1} \left(n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[\frac{\pi}{N_2} \left(n_2 + \frac{1}{2} \right) k_2 \right] \quad (7)$$

4 Classification and Identification

These two stages each deal with one of the topics of classification and identification.

4.1 Classification

In the previous part, we showed some coin-specific characteristics which can be used to tag a coin image. This part describes how to make use of the features in order to decide upon the class of a coin in a clear way.

The fingerprint of an image has the following composition:

- five distance distribution vectors (for 2, 4, 8, 16 and 32 bins)
- a Fourier transform of the magnitude vector over the angle distribution for 240 bins
- first 8 components of the DCT over the L-channel
- first 4 DCT components for the U-channel
- first 4 DCT components for the V-channel.

By calculating the nearness between the images (through the calculation of a distance value), the similarity can be checked. Logically, the best solution for this purpose is to make use of the Euclidian distance between two points in an N-dimensional space, obtained by using the following formula:

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}. \quad (8)$$

4.2 Identification

The identification has two steps:

At first, to preselect a group of images which have similar texture; we have to use the DCT method for fingerprinting.

Then, the distance and angle distribution method is used to identify three best matches from the group preselected at the first step.

5 Results

The experiments were performed on a Core2DUO machine with 2GB of RAM, Windows 7 OS, 64 bit.

We tested a set of about 120 free coin pictures taken from the web. So the images were acquired by various persons which were using different methods for capturing them (regarding of illumination and rotation), also with different sizes and dpi.

The process of determining the characteristics of a coin is fast, with an average duration of 13.8 milliseconds. Nevertheless, as it's been said before, because of heterogeneity of the input material, the period of time could go as low as 9 milliseconds, or even as high as 35 milliseconds.

The characteristics of these reference coins were stored in a database, in order to be used to match against another set of test coins, completely different. Due to the relatively modest coin collection available, the bulk of the processing was allocated to feature extraction of the target coin as opposed to matching against reference coins. For a real coin catalogue, it is expected that most of the effort would be put into comparing the coin against every data stored in the database, which could be easily parallelized.

The experiments made on modern coins were very promising with fine results in respects of accuracy. The wrong classifications were obtained only because of the poor image quality (like a very dark picture).

It was expected that tests performed on medieval or older coins were not to be as convincing as those made on modern ones, but it has to be taken into consideration the fact that there are some problems associated with pre-modern coins [11,12]: they are not a result of a factorial process, the variety in the stamps of medieval coins is inferior to the variety in stamps of modern coins and they are often strongly eroded because of the frequent use and passing of time.

6 Conclusions

This paper meant to show a method for automatic classification of coin pictures using two effective characteristic types, a method which obtained very good results on modern coins. The classification of old coins proved not to be as efficient as for the modern ones. Furthermore, our work showed an approach for dealing with photo capture weaknesses (e.g. illumination problems) thus the different lighting conditions, the various shiny parts due to specular reflection are corrected up to a certain degree so that the final classification can contain the least amount of errors as possible.



Fig. 9. Coins that usually tend to be correctly classified when artificial rotations and lighting effects are applied.



Fig. 10. Coins that are sometimes misclassified, mostly due to their shape inconsistencies and surface defects.

In Fig. 9 is presented a set of coins that are usually correctly classified. The coin fingerprinting mechanism in case of this set seems to work fine, thus remaining unaltered when artificial rotations and post-processing lighting conditions are applied.

In Fig. 10 is presented a set of coins with classification problems. It can be observed that most of the problems are generated by the advanced state of deterioration of the reference coins thus any post-processing algorithms are not sufficient to restore the individual coin fingerprints to a reasonable level.

The results may be considered satisfactory in the end due to the obvious deterioration of the misclassified coins and having in mind that some visibly deteriorated coins are usually classified correctly. However it is important to say that the database size is rather small and may not be considered representative for large batch classifications. Having much more coin classes to classify will obviously result in a smaller average distance between the different coins fingerprints, thus making the process more sensitive to errors.

On the other hand when dealing with a small number of classes, even in the case of large input batches the presented approach should lead to good results for both the classification quality and the processing speed.

7 Future Work

The next steps of the work should be concentrated in solving the known problems of the system, but we have to mention that a lot larger input database will decrease the actual percentage of correct identification and classification.

Another important aspect of the future work should refer on improvements for the case of medieval coins classification. It will be implemented a system that uses images of coins and creates a pyramid of different sizes of the original image.

Each of the system components has a package (a set of unique up-sampling – down-sampling filters) that is used to recursively construct a pyramid.

The operations involved in generating a new level are:

1. down-sampling the image of the current level, gaining an image half its length and half its width
2. up-sampling the same image, obtaining an approximation of the original image
3. obtaining a residual image by extracting the differences between the original image and the up-sampled one
4. using the residual image in fingerprinting and classification next to some of the current fingerprints

Fig. 11 stands for an example of the resulting pyramid. When a cycle ends, the down-sampled image becomes a new layer in the pyramid. The processing goes on by creating more levels until a small enough image is generated. The limit is a linear image, with either only one row or only one column of pixels.

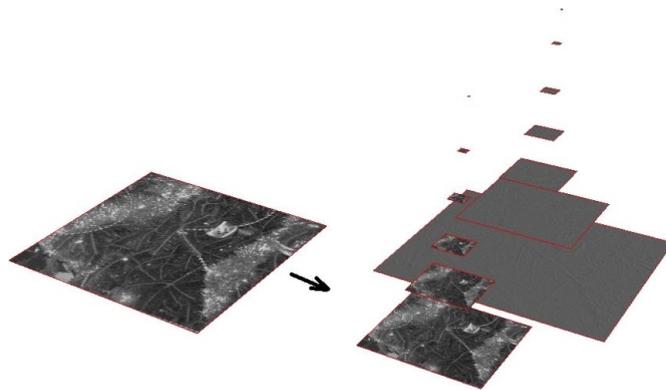


Fig. 11. Constructing a pyramid of different image resolutions

There is a possibility that the system of pyramidal coding offers better signatures and faster classification [21]. Some candidates could be rejected from lower pyramidal levels where signatures are easier to compute and to compare one against an entire dataset).

Another fact which can be considered an advantage is that lower levels are not very sensitive to noise, illumination and degradation of the coin.

Acknowledgements

The authors want to show their appreciation to Andrei Tigora, Mihai Zaharescu and Alexandru Gradinaru for their great ideas, support and assistance with this paper.

References

1. Adameck, M., Hossfeld, M., and Eich, M. (2003). Three color selective stereo gradient method for fast topography recognition of metallic surfaces. *Machine Vision Applications in Industrial Inspection*, 11, 128-139, DOI: 10.1117/12.473968.
2. Bag, S., Bhowmick, P., Behera, P., and Harit, G. (2011). Robust binarization of degraded documents using adaptive-cum-interpolative thresholding in a multi-scale framework, In *Proceedings of International Conference on Image Information*, 1-6.
3. Bourgin, D. (2009). Color spaces FAQ, Available from: [http:// www.ilkeratalay.com/ colorsfaq.php](http://www.ilkeratalay.com/colorsfaq.php). [Accessed 1st November 2013].
4. Canny, J., (1986). A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6), 679-714.
5. Chai, H.Y., Wee L.K., Supriyanto, E. (2011). Edge detection in ultrasound images using speckle reducing anisotropic diffusion in canny edge detector framework. In *Proc. 15th WSEAS international conference on Systems*, 226-231
6. Davidsson, P., (1996). Coin classification using a novel technique for learning characteristic decision trees by controlling the degree of generalization. *Ninth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/ AIE-96)*, 403-412.
7. Deriche, R., (1987). Using Canny's criteria to derive an optimal edge detector recursively implemented. *International Journal of Computer Vision*, 1(2), 167-187.
8. Fukumi, M., Omatu, S., and Kosaka, T. (1992). Rotation invariant neural pattern recognition system with application to coin recognition. *IEEE Transactions on Neural Networks*, 3(2), 272-279.
9. Green, B. (2002). Canny edge detection tutorial, Available from: [http:// dasl. mem. drexel. edu/ alumni/ bGreen/ www. pages. drexel. edu/ _weg22/ can_tut. html](http://dasl.mem.drexel.edu/alumni/bGreen/www.pages.drexel.edu/_weg22/can_tut.html). [Accessed 23rd November 2013].
10. Jianzhuang, L., Wenqing, L., and Yupeng, T. (1991). Automatic thresholding of gray-level pictures using two-dimension Otsu method, In *Proceedings of International Conference on Circuits and Systems*, 325-327.
11. Maaten, L.J.P. and Boon, P.J. (2006). COIN-O-MATIC: a fast and reliable system for coin classification. In *Proceedings of the MUSCLE Coin Workshop 2006*, Berlin, Germany, 7-17.
12. Maaten, L.J.P. and Postma, E.O. (2006). Towards automatic coin classification, In *Proceedings of EVA-Vienna*, 19-26.
13. Menon, V., Babbar, U., and Dasguta U. (2009). Web-based image similarity search, unpublished.
14. Otsu, N., (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62-66.
15. Quinlan, J.R., (1986). Induction of decision trees. *IEEE Trans. Machine Learning*, 1(1), 81-106.
16. Rangarajan, S. (2005). Algorithms for edge detection, Available from: [http:// www.ee.sunysb.edu/ ~cvl/ ese558/ s2005/ Reports/ Srikanth%20Rangarajan/ submission.doc](http://www.ee.sunysb.edu/~cvl/ese558/s2005/Reports/Srikanth%20Rangarajan/submission.doc). [Accessed 3rd November 2013].
17. Sung, T.-Y. (2007). Memory-efficient and high-performance 2-D DCT and IDCT processors based on CORDIC rotation, In *Proceedings of 7th WSEAS International Conference on Multimedia Systems & Signal Processing*, Hangzhou, China, April 15-17, 7-12.

18. Tică, Ș.N., Boianjiu, C.-A., and Tigora, A. (2013). A method for automatic coin classification. In Proceedings 1st WSEAS International Conference on Image Processing and Pattern Recognition, Budapest, Hungary, December 10-12, 188-198.
19. Tigora, A. (2013). An image binarization algorithm using watershed-based local thresholding. In Proceedings of 1st WSEAS International Conference on Image Processing and Pattern Recognition (IPPR '13), Budapest, Hungary, 154-160.
20. Zaharescu, M. and Petrescu C.I. (2013). Edge detection in document analysis, *Journal of Information Systems, Operations Management*, 7(1), 156-165.
21. Zhang, W. and Bergholm, P. (1997). Multi-scale blur estimation and edge type classification for scene analysis, *International Journal of Computer Vision*, 24(3), 219-250.
22. Alessandro A., Francesco B., Dmitry C. (2011), A Survey of Specularity Removal Methods, *COMPUTERGRAPHICS forum*, Volume 30 (2011), number 8 pp. 2208–2230