Published in the Proceedings of the 4th *International Workshop on Visual Form*, May 28-30, Capri, Italy (Springer-Verlag, series "Lecture Notes in Computer Science", number LNCS 2059).

Qualitative Estimation of Depth in Monocular Vision

Virginio Cantoni, Luca Lombardi, Marco Porta, and Ugo Vallone

Dipartimento di Informatica e Sistemistica, Università di Pavia
Via Ferrata,1 – 27100 – Pavia – Italy
{cantoni, luca, porta, vallone}@vision.unipv.it

Abstract. In this paper we propose two techniques to qualitatively estimate distance in monocular vision. Two kinds of approaches are described, the former based on texture analysis and the latter on histogram inspection. Although both the methods allow only to determine whether a point within an image is nearer or farther than another with respect to the observer, they can be usefully exploited in all those cases where precision is not critical or single images are the only source of information available. Moreover, combined with previously studied techniques, they could be used to provide more accurate results. Step by step algorithms will be presented, along with examples illustrating their application to real images.

1 Introduction

Monocular vision concerns the analysis of data obtainable from single images. While in stereo or trinocular vision spatial information can be drawn by comparing different images of the same scene, in monocular vision the analysis can be performed only by studying intrinsic characteristics of the representation. For example, comparisons and statistical investigations can be carried out on the distribution of gray levels (in the monochromatic case) or of the red, green and blue channels (in the RGB case) of the pixels composing the image. Therefore, results which can be obtained are influenced by the acquisition systems employed and by the spatial and tonality resolutions adopted.

The two techniques we propose in this paper are based on texture and histogram analysis. Although only qualitative information can be obtained through them, we think they can be anyway useful when the precision of the results is not critical or when there is no other source of data available. Monocular vision may be notably more advantageous than techniques exploiting couples or sequences of images, since it requires simpler acquisition systems and is computationally more efficient in terms of execution times. Moreover, qualitative estimations could be used to confirm evaluations obtained by means of other more precise techniques (such as binocular vision, infrared sensors, etc.).

The paper is structured as follows. Section 2 will describe the approach based on texture analysis. After a brief discussion about previous works regarding the topic and an introduction to the theory behind it, two practical algorithms will be presented, of which the second can be used to distinguish between nearer and farther zones within

images in perspective projection. Section 3 will describe the technique based on histogram analysis. A brief introduction will precede the presentation of the implemented algorithm, which, like the texture-based one, allows relative distances within images in perspective projection to be estimated. Section 4, lastly, will draw some conclusions and suggest directions for future research. All the algorithms are followed by examples illustrating their use.

2 Texture Analysis as a Source of Spatial Information

In general, a surface can be considered as being characterized by some form of *texture* if the relevant shapes composing it are uniformly distributed, i.e. if they do not differ very much in appearance, size and density throughout the extension of the surface itself [1].

Essentially, two different approaches have been proposed for texture analysis. Some researchers (such as [2]) follow a *structural* approach, which requires the real structure of texture to be determined (periodicity, uniformity, symmetry, etc.). Although this is probably the method used by the human vision system to infer 3D structure of the environment, it is difficult to automate. Other more feasible techniques limit themselves to making assumptions about the texture arrangement. For instance, if the texture is isotropically distributed, i.e. line segments composing the real surface have not a prevalent direction, three-dimensional information can be obtained from the "preferred" direction observed. This approach was first proposed by Witkin [3] and subsequently improved by Davis et al. [4]. G♫rding [5] and Blake et al. [6] perfected the method under the hypothesis of orthographic projection. As regards perspective projection, Kanatani [7] provided a rigorous mathematical description of the problem. Under the hypothesis that texture is composed of points and straight lines only, he developed a technique based on texture homogeneity and density. However, this algorithm is not able to produce good results applied to real scenes. In fact, Kanatani analyzes how the density of image points varies as distance increases, asserting that for greater distances density along a surface grows. Unfortunately, in most real cases the number of relevant points decreases, because of lenses' blur.

We will now concentrate on Witkin's algorithm, in the form corrected by Davis et al. [4]. In Section 2.2 it will be used to determine, given two points on an image in perspective projection, which one corresponds to a farther region in the real scene.

2.1 Texture Analysis in Orthographic Projection

When observing shapes on a plane surface, two different kinds of geometric distortions can be noted: (1) as the surface departs from the observer, shapes appear to be smaller and smaller; (2) the more a surface is inclined with respect to the image plane, the more shapes on the image appear to be flattened towards the tilt direction (*foreshortening* effect). As will be shown, such distortions can be usefully exploited to get spatial information about the real scene.

In image processing, when the *orthographic projection* hypothesis is satisfied, which means that every point in the three-dimensional space is orthogonally projected

on the image plane (see Figure 1), effect (1) can be ignored and it becomes easier to estimate the orientation of the plane on which the real scene lies.

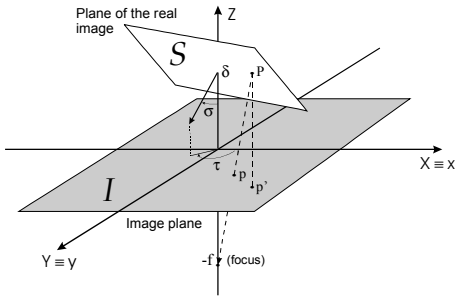


Fig. 1. Reference cartesian system. Orthographic (point *p'*) and perspective (point *p*) projections of the spatial point *P* on the image plane

As already stated, for orthographic projection Witkin [3] was the first to face the problem of drawing the spatial arrangement of a plane from image texture analysis, under the hypotheses of *independence* and *isotropy*, which can be summarized as follows:

1. Given an image on the image plane, the possible orientations of the planes on which the original scene may lie are equiprobable.
2. On the plane on which the original image lies, the possible orientations of the tangents to the curves composing the image itself are equiprobable.
3. Orientations of the elements in the real image are statistically independent.

Witkin's studies led to a practical algorithm by which it is possible to estimate the σ and τ polar coordinates (slant and tilt) of the plane on which a scene lies (Figure 1). Let's suppose to have a plane S, on which there are curves and shapes satisfying the isotropy and independence hypotheses. Also, suppose the orthographic projection of S on the image plane I is satisfied. Then, directions β of the tangents to the curves on S will have $p.d.f.(\beta) = 1/\pi$ as probability density function, where $\beta \in [0, \pi[$. As regards parameters σ and τ, the *p.d.f.* function can also be expressed as $p.d.f.(\sigma, \tau) = (\sin\sigma)/\pi$. In fact, consider the *gaussian sphere*, which is the unit radius sphere formed by all the possible normals to all the possible plane arrangements in space. For each value of σ, the possible values of τ define a circumference on the gaussian sphere, whose radius approaches 1 as σ approaches $\pi/2$. The probability that the orientation of a normal corresponds to a certain point on a certain circumference is the same for every point on it and is proportional to the length of the circumference itself ($2\pi \sin\sigma$). Therefore, the *p.d.f.* function can be simply expressed in function of $\sin\sigma$, without the need to introduce parameter τ. Since, in general, it is not possible to measure β angles directly, it is necessary to find their transformation into the corresponding angles on the image plane I (let's call them α). If the orientation of plane S is (σ, τ), the image on I can be translated into the one on S by rotating it by the same angles. Then, supposing $\tau = 0$, point $p(x, y, 0)$ on I will have $X = x\cos\sigma$, $Y = y$ and $Z = x\sin\sigma$ as coordinates on S. Since the orthographic projection of a spatial point (X, Y, Z) on I is (X, Y), a sim-

ple way to get the projection of a curve which lies on plane S on plane I consists in "placing" the curve on I, rotating it by (σ,τ), and projecting it again on I. Now, let's consider versor $t = [cos\beta, sin\beta]$, tangent to a curve on S in a certain point, according to an angle β. After the rotation on I, it will become $t^* = [cos\beta cos\sigma, sin\beta]$. Therefore, we have that $tan\alpha = sin\beta/(cos\beta \cdot cos\sigma) = tan\beta/cos\sigma$, where α is the orientation of versor t^* on plane I. If $\tau \neq 0$, it is sufficient to add it to α, which means that $\alpha = atan(tg\beta/cos\sigma) + \tau$ and $\beta = atan(cos\sigma \cdot tan(\alpha - \tau))$.

Considering that, in general, $p.d.f.(\varphi(x)) = p.d.f.(x) \cdot dx / d\varphi(y)$, we find:

$$p.d.f.(\alpha(\beta) \mid \sigma,\tau) = p.d.f.(\beta \mid \sigma,\tau)\frac{\partial\beta}{\partial\alpha} = \frac{1}{\pi}\frac{cos\,\sigma}{cos^2(\alpha-\tau) + cos^2\,\sigma \cdot sin^2(\alpha-\tau)}$$

However, an image will be composed of many curves and hence it is also necessary to find the compound probability density function $p.d.f.(A \mid \sigma,\tau)$, where A is a set $\{\alpha_i\}$ containing the various α angles measured on the image plane for all the curves on it. If all α_i are independent, the following can be obtained:

$$p.d.f.(A = \{\alpha_1,...,\alpha_n\} \mid \sigma,\tau) = \prod_{i=1}^{n} p.d.f.(\alpha_i \mid \sigma,\tau) = \tag{1}$$

$$= \prod_{i=1}^{n} \frac{\pi^{-1}\,cos\,\sigma}{cos^2(\alpha_i-\tau) + cos^2\,\sigma \cdot sin^2(\alpha_i-\tau)}$$

where n is the number of α angles. Substantially, the preceding expression allows us to calculate how much probable a set of values $\{\alpha_i\}$ is, given σ and τ. However, we are interested in the opposite problem, i.e. in finding the probability to have a particular couple (σ,τ) given a set of values $\{\alpha_i\}$. Applying the Bayes rule and normalizing so that the integral of the probability density function is equal to one, we have:

$$p.d.f.(\sigma,\tau \mid A) = \frac{p.d.f.(\sigma,\tau)\,p.d.f.(A \mid \sigma,\tau)}{\iint p.d.f.(\sigma,\tau)\,p.d.f.(A \mid \sigma,\tau)\,d\sigma\,d\tau} \tag{2}$$

That couple (σ^*,τ^*) which maximizes $p.d.f.(\sigma,\tau \mid A)$ is the more probable one and therefore it can be assumed as the orientation of plane S (*maximum likelihood estimate* technique). From expressions (1), and remembering that $p.d.f.(\sigma,\tau) = sin\sigma/\pi$, the value of the numerator of expression (2) can be calculated as follows:

$$p.d.f.(\sigma,\tau)\,p.d.f.(A \mid \sigma,\tau) = \frac{sin\,\sigma}{\pi}\prod_{i=1}^{n}\frac{\pi^{-1}\cos\sigma}{\cos^2(\alpha_i-\tau) + \cos^2\sigma \cdot sin^2(\alpha_i-\tau)} = \tag{3}$$

$$= exp\left(log\left(\frac{sin\,\sigma}{\pi}\right) + \sum_{i=1}^{n} a_i\, log\frac{\pi^{-1}\,cos\,\sigma}{cos^2(\alpha_i-\tau) + cos^2\,\sigma \cdot sin^2(\alpha_i-\tau)}\right)$$

where a_i is the number of measures whose value is α_i.

The previous considerations lead to the following algorithm to estimate slant and tilt of the plane on which a scene lies.

Algorithm 1. The practical algorithm we have implemented is derived from that of

Witkin and is composed of the following steps:

1. If the original image on the image plane is known to be affected by noise, it is filtered through a low-pass filter.
2. After having been normalized, the image undergoes an edge-detection process, through the Sobel operator. As a result, two different "images" are obtained from the gradient, namely the module image and the phase image.
3. The module image is thresholded against a certain value, to get a binary image (mask) identifying only the most relevant edges. We are studying for an automatic evaluation of this value.
4. For each pixel in the original image whose value in the mask is one, the direction of its tangent is calculated. This is done by simply rotating by 90 degrees the corresponding value in the phase image.
5. Interval [0, π[is subdivided into n sub-intervals. An array $A = \{a_1,...,a_n\}$ is built in which a_i is the number of values obtained in step 4 falling into the i^{th} sub-interval (α_i). Interval [0, π/2] (σ values) is subdivided into m sub-intervals and interval [0, π[(τ values) is subdivided into p sub-intervals.
6. For each possible couple (σ_j, τ_k) ($j \in [0, m-1]$, $k \in [0, p-1]$), expression (3) is calculated. That couple (σ_{j*}, τ_{k*}) for which the result is maximum is taken as the orientation of the plane containing the real scene in the three-dimensional space.

Experimental Results. We tested algorithm 1 with many images and the results obtained confirm its effectiveness in estimating slant and tilt of real scenes. Figure 2 shows three example images, for which the computed σ and τ are reported. Of course, the algorithm considers them as if they were in orthographic projection. To give an intuitive insight of the results obtained, circles oriented according to the calculated values are superimposed on the original images, and, at the center of the circles, segments normal to the estimated planes are placed. For the algorithm, the following parameter values have been assumed: $n = 64$, $m = 90$, $p = 180$, $\alpha_i = \pi(1/2+i)/n$, $\sigma_j = \pi(1/2+j)/2m$, $\tau_k = \pi(1/2+k)/p$, where $i \in [0, n-1]$, $j \in [0, m-1]$ and $k \in [0, p-1]$.

$\sigma = 56.5°$	$\sigma = 69.5°$	$\sigma = 58.5°$
$\tau = 90.5°$	$\tau = 147.5°$	$\tau = 87.5°$

Fig. 2. Three examples of application of Algorithm 1. Beneath each image, which is considered as if it was in orthographic projection, the calculated σ and τ parameters are reported

2.2 Texture Analysis in Perspective Projection

In perspective projection, each point in the three-dimensional space is projected on the image plane through the focus of the optical system (see again Figure 1). When, as often occurs, the orthographic projection hypothesis is not satisfied, the algorithm presented in the previous section can only produce approximated results. In fact, shape size lessening and the foreshortening effect cause the image on the image plane to be less "uniform", thus weakening the concept of texture itself. However, intuitively, if from an image in perspective projection sufficiently small regions are extracted, it will be reasonable to consider them as if they were in orthographic projection. If algorithm 1 is then applied to each of them, we will find that those regions which are farther with respect to the tilt direction produce greater values for parameter σ, because shapes turn out to be more flattened. In fact, referring to Figure 1, given a point *P(X,Y,Z)* on plane S, the corresponding coordinates on the image plane will be $x = fX/(f+Z)$ and $y = fY/(f+Z)$. Let's suppose $\tau = 90°$ and consider the perspective projection of a circle placed on S on the image plane. The resulting image, unless $\sigma = 90°$, will be an ellipse, as shown in Figure 3.

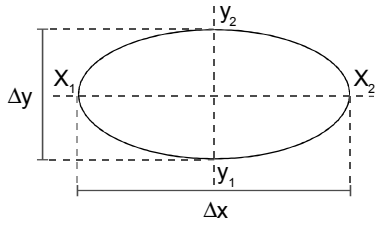


Fig. 3. Projection of a circle on the image plane

If the distance of point y_1 from the image plane is Z and the circle's diameter is D, the ellipse's Δx width can be calculated as $fD/[f+Z+(D/2)\sin\sigma]$. If y_1 has Y as the corresponding coordinate in space, the ellipse's Δy height is $\Delta y = y_2 - y_1$, where $y_1 = fY/(f+Z)$ and $y_2 = f(Y+D\cos\sigma)/(f+Z+D\sin\sigma)$. Since the equation of plane S is $Z = \tan\sigma \cdot Y + \delta$ (where δ is the coordinate of the intersection of plane S with the Z axis), we obtain: $\Delta y = y_2 - y_1 = fD\cos\sigma(f+\delta)/[(f+Z)(f+Z+D\sin\sigma)]$. If the projection was orthographic instead of perspective, relation $\cos\sigma_o = \Delta y/\Delta x$ would hold (where σ_o indicates just the slant calculated in the orthographic case). In fact, Δx would be equal to D and Δy would be the projection of D on the image plane, i.e. $\Delta y = D\cos\sigma_o$. However, if a sufficiently small area is selected on the image to be processed, algorithm 1 can provide a good approximation of the image slant. Then, supposing this is the case, from the previous expressions we have:

$$\cos\sigma_o = \frac{\Delta y}{\Delta x} = \frac{\cos\sigma_p (f+\delta)\left(f+Z+\frac{D}{2}\sin\sigma_p\right)}{(f+Z)(f+Z+D\sin\sigma_p)}$$

where σ_p indicates the slant calculated in the perspective projection case. It is evident that when Z increases (i.e. the distance from the observer grows) also σ_o must increase.

The previous considerations suggest a simple but effective way to take into account perspective when trying to estimate distance in images for which the orthographic projection hypothesis is not valid.

Algorithm 2. Given an image on the image plane, to determine whether a certain point is farther or nearer than another (with respect to the observer), it is sufficient to consider two areas around them and apply algorithm 1. The farther point is that for which σ is greater.

Experimental Results. Figure 4 shows some examples of application of algorithm 2. Rectangles on the images identify the regions analyzed through algorithm 1, whose estimated σ and τ are reported. Values assumed for the parameters are the same as for the examples in Section 2.1 (n = 64, m = 90, p = 180).

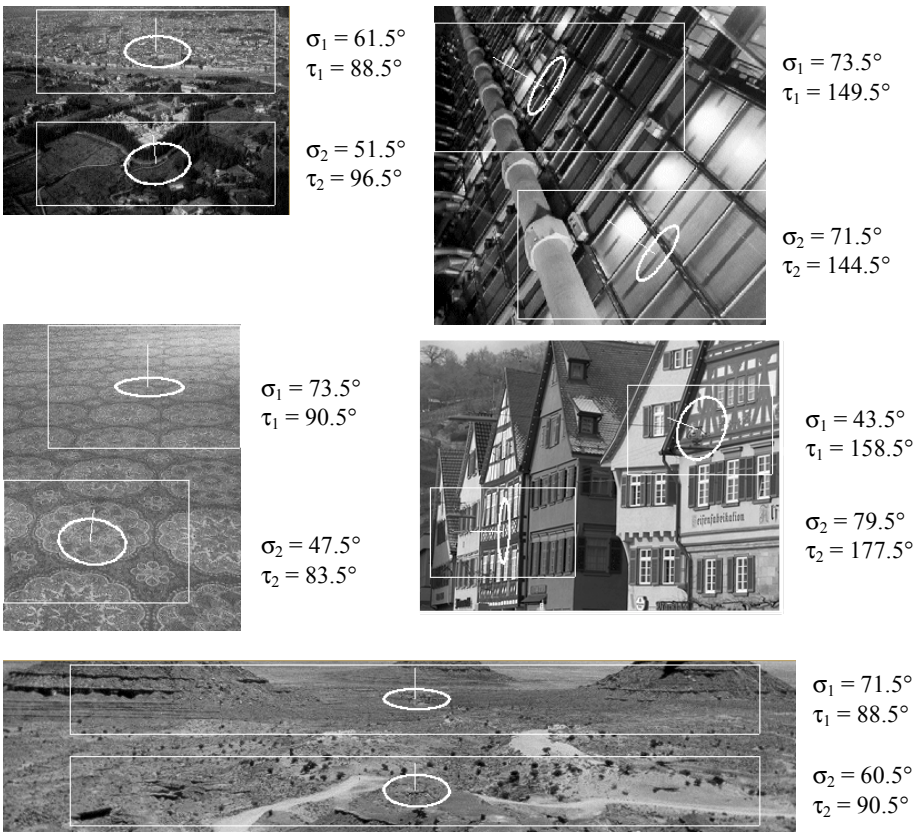


Fig. 4. Examples of application of Algorithm 2. At the right of each image, in correspondence with rectangles identifying the examined regions, the estimated σ and τ parameters are reported

3 Histogram Analysis for Distance Estimation

In this section, we will present another algorithm for qualitative estimation of distance in monocular vision: histogram analysis. Instead of using geometric parameters, it exploits physical properties of the medium in which the objects in the images to be analyzed are contained. The technique, which is primarily intended for being used in combination with other methods, is currently in a development stage and we are still working on it to automate the choice of the values for some key parameters.

Everyone can note that, for instance in images representing landscapes, edges of distant elements (such as mountains) are not very clear: colors of different objects tend to blend each other and generate zones with nearly uniform colors. Such an effect is more marked if haze or fog is present, or when underwater images are examined. In general, it is more perceivable in images with very long depth of field compared with the medium opacity.

Rays of light in an opaque medium are diffused by its molecules. For example, air contains a great number of water particles, which refract light. In a monochromatic image, the background's gray levels tend to be the weighed average of all the gray levels present in the image itself. As a result, the image becomes more and more uniform as the distance from the observer increases, thus producing a sort of blur. This blur, however, is not to be mistook for that produced by camera lenses, which must be absolutely avoided to get valid results from the technique we are now going to describe.

3.1 The Algorithm

As already stated, the zones of an image which are perceived as more blurred (i.e. which are farther with respect to the observer) have gray levels gathered around an average value. On the contrary, in those areas where edges are sharper (i.e. which are nearer with respect to the observer) gray levels are more scattered. Information about the distance of a zone with respect to another can thus be obtained by analyzing the gray level spectrum in these regions.

The algorithm we present here is based on variance applied to the image histogram.

Algorithm 3. Consider two points P_1 and P_2 on an image. To determine which one is nearer (or farther) with respect to the observer, we can proceed according to the following steps:

1. Two areas A_1 and A_2 around P_1 and P_2 are extracted from the image and their histograms are obtained.
2. Both for A_1 and A_2, the average gray levels x_{a1} and x_{a2} of their pixels are calculated. In general, indicating with $f(x)$ the number of pixels whose gray level is equal to x, the average is obtained as follows:

$$x_a = \frac{\sum_i x_i f(x_i)}{\sum_i f(x_i)}$$

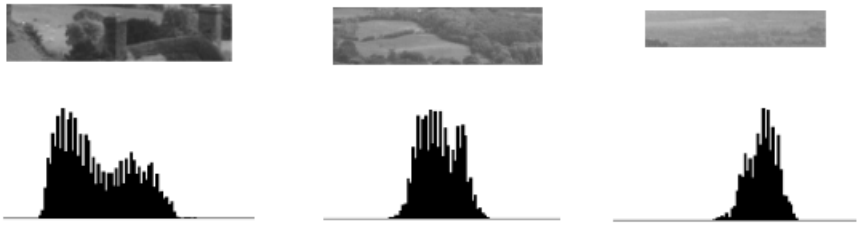
3. Both for A_1 and A_2, variances σ_1^2 and σ_2^2 are calculated in the following way:

$$\sigma^2 = \frac{\sum_i (x_i - x_a)^2 f(x_i)}{\sum_i f(x_i)} = \frac{\sum_i x_i^2 f(x_i)}{\sum_i f(x_i)} - x_a^2$$

4. The point which is farther with respect to the observer will be that whose variance is lower.

Experimental results. As an example of application of algorithm 3 to a real image, Figure 6 displays the histograms relative to the three areas highlighted by rectangles (a, b and c) in Figure 5.

Fig. 5. An example of application of algorithm 3. White rectangles are used to highlight the areas analyzed (a, b and c)



σ = 13.99 σ = 7.69 σ = 5.75

(a) **(b)** **(c)**

Fig. 6. Histograms relative to the areas highlighted in Figure 5, and corresponding variances

As can be noted, the histogram of the nearest zone (rectangle *a*) is more scattered than that of the farthest zone (rectangle *c*) and, accordingly, the variance of the first area (13.99) is greater than that of the second one (5.75).

Of course, dimensions of rectangles may remarkably affect the results. We are currently trying to make such a choice automatic, based on a preliminary segmentation phase to select sufficiently homogeneous areas.

4 Conclusions

In this paper we have presented two techniques which allow distances to be estimated from a relative point of view. That is, they allow to determine whether a point within an image is farther or nearer than another with respect to the observer.

Two algorithms, applicable to images in perspective projection, have been described. The first (Algorithm 2), based on texture analysis, has been tested with a great number of real images in which some form of texture was recognizable, always giving very good results. The second (Algorithm 3), which is based on histogram analysis and is still to be perfected, has shown its good applicability especially to representations with long depth of field (e.g. landscapes). Although some precautions are to be taken (there must not be any form of lens blur), the use of this method is favored by its computational lightness.

We hold that, although they cannot be exploited for actual distance estimations, the proposed algorithms, integrated with others previously studied [8], can be usefully used for qualitative evaluations. Moreover, they can be exploited as a preprocess step able to detect regions of interest for successive application of more expensive techniques, to provide more accurate results. Current work is devoted to this aspect and to the integration of the system with different sources of information.

References

1. Gibson, J.: The Perception of the Visual World. Houghton Milfin, Boston, MA (1950).
2. Blostein, D., Ahuja, N.: Shape from Texture: Integrating Texture-element Extraction and Surface Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (1989) 325-342.
3. Witkin, A.P.: Recovering Surface Shape and Orientation from Texture. Artificial Intelligence, 17 (1981) 17-45.
4. Davis, L.S., Janos, L., Dunn, S.M.: Efficient recovery of shape from texture. IEEE Transactions on Pattern Analysis and Machine Intelligence, 5 (1983) 485-492.
5. G♬rding, J.: Shape from Texture and Contour by Weak Isotropy. Artificial Intelligence, 64 (1993) 243-297.
6. Blake, A., Marinos, C.: Shape from Texture: Estimation, Isotropy and Moments. Artificial Intelligence 45 (1990) 323-380.
7. Kanatany, K., Chou, T.: Shape from Texture: a General Principle. Artificial Intelligence, 38 (1989) 1-48.
8. Matessi, A., Lombardi, L.: Vanishing Point Detection in the Hough Transform Space. Proceedings of the Fifth International Euro-Par Conference, Tolouse, France, August/September (1999) 987-994.