Renderings. Some technical and non technical questions raised by the use of computers in the visual analysis of architecture.

Javier Monedero, Architect, Ph.D
Professor E.T.S. Architectura, Barcelona
Universitat Politecnica de Catalunya

It should be expected, in a Congress, that participants bring with them, not only ideas, but also results or, at least, stimulating images. In the Laboratory of Architectural Graphic Techniques at the ETS of Barcelona, we have spent some time generating images directly related with architecture, based on the work of both students and professors. These images have been produced with academic purposes, but also in relation with some works carried out with City Institutions interested in the study of the evaluation of environment visual impact and the role that computers may play in this area. In our previous Congress, in Barcelona, we showed some of these images [1], obtained by direct digital processing of bitmaps. In another Congress, later [2], we showed some other images, obtained by rendering, with simple local models (Phong models) and some tricks that helped to make them more realistic.

Although I do agree with the old Chinese saying that a good image is worth a thousand words, in this case, I have thought more convenient to present a paper that may be read quietly by those interested in these subjects, that might be useful just as it gathers references known by many but grouped in a particular order, and that pretends, respectfully, to criticize the actual situation. This can explain why we consider that the results we have obtained should be improved by new and better techniques and why we think that this dissatisfaction should be shared by others who do not seem to feel the same as we do. The aim of this contribution is, therefore, to reflect on the actual situation and the ways there seem to be open for us to follow.

1. What is the use of color?

From Galileo to Descartes, science has promoted ways of knowledge that helped us to go "beyond appearance" and could "shed light on" the "objective" truths that remain covered by deceptive veils. Of course, this healthy attitude is at the ground of the achievements our civilization feels, rightly, more proud of. The problem is that, as it happens with all conquest, the conqueror is tempted to extend his domain farther away from what it is reasonable. And, certainly, it is not reasonable to speak of "deceptive" appearances in a field in which the material is provided, precisely, by appearances.

What is the use of color? From my point of view, this question is so absurd when it is directed to one of the three great visual arts (and one essential part of architecture, which distinguishes it from building science, consists of its being a visual art) that it could only be responded by a similar question : what is the use of architecture?

But so it happens that this prejudice about color reaches directly architects and schools of architecture. Many of them, as we all know, are trying since three centuries ago to appear as respectful as engineers and not to appear as frivolous as decorators. And so, the representation of what was called, in the empiricist tradition, from Locke, "secondary qualities", has received a minimal attention in occident schools of architecture. Color, texture, modes of illumination, and material qualities have been regarded with mistrust by a line of thought that, from renaissance, has considered Form as the "primary quality", steady, regular, universal, apt to be object of knowledge.

That this line of thought has come to he considered the norm in our schools, despite the fact that it implies a ninety degrees turn in the course of European thought, a radical turn that separates XV century from what we call "Middle Ages", testimonies how deep it is rooted in our minds. But this, that should be considered as a sign of progress from a scientific point of view, cannot be considered but as change of taste from the point of view of the visual arts. It is a fact that, with more limited means, the painters and the architects that created the medieval madonnas, the glasswork of Chartres or the Venetian Palazzi, had a better trained sensitivity to color that what has been showed by the great majority of their famous successors.

Our schools of architecture have missed this tradition. But is interesting to note that the exceptions to this rule have constituted themselves, precisely, as the most important references to their generation. The Beaux Arts in the XIX century, and the Bauhaus in the XX, remain as the two notable exceptions to this rule, in what regards the importance given to color, and the importance given to direct work with materials, in the education of an architect.

The lost of this tradition is the cause of the scarce attention that progress in the field of Rendering in Computer Graphics is receiving in our schools, and the passive acceptation of some techniques that shall be used but, above all, should be seriously criticized, at the present moment, as a contribution to their development.

What follows is, in this line, an evaluation of the sate of the art from the point of view of architecture and a presentation of a general position: once we have passed an epic phase in which everything was justified because new ways of working with light and colors and materials have appeared, thanks to the effort of a few brilliant researchers, and, now, in a moment in which too many students have started to incorporate these new methods to their projects with, light, color and, why not, music, it is urgent to ask for some visual reeducation. These are results which are admirable for the industry and cleverness of the people that have develop them; but that cannot be accepted, in general, from the point of view of visual quality in architecture.

2. Computer lights

Rendering software utilizes currently 4 types of lights : "ambient", "point", "direct" and "spot". We have all get used to these words which have become the common jargon of anyone working in this field. But, when such a radical abstraction as it implied by these words, becomes a habit, the moment has come to remember what was abstracted and to consider whether this abstraction is still useful or, on the contrary, it can be, not only insufficient but, what is worst, an unacceptable distortion.

What is an "ambient" light? The answer is: a light placed nowhere, that comes from every direction with equal intensity. But this is, obviously, an scheme very far away form any real case we might think of. Outside, in a cloudy day, we can find something similar to an "ambient" light. Inside, in a room with artificial lights everywhere, floors, walls and ceilings of the same color and reflectivity, we might also find something similar. But such an interior is a rather rare interior. The only one I can think of at the moment, that resembles something remotely similar, is the inside of the aeronave at the beginning of Stanley Kubrick's film "2001. A space Odyssey." : a quite disturbing space, just because of an unusual illumination coming from nowhere, absolutely unfamiliar to us and rather difficult to produce.

What do we call a "point" light? Something that neither exits nor can exist in a real environment. Not just because there are no lights condensed in a single point (this could be an acceptable abstraction in some particular cases) but, mainly, because there are no lights emitting their rays in every direction except if they tloat; at the minimum, they cannot send rays in the direction of their support. The majority of lights we know incorporate some kind of lampshade and, apart from this, they do not emit equally in all directions but, rather, they follow an intensity distribution that usually is provided in the shape of goniometric diagrams by their producers.

What do we understand by a "direct light? The answer might suggest that it is a more consistent artifact than the previous ones. On condition that we forget indoors illumination (no current artificial light will throw parallel rays) and on condition that we forget the possibility of dealing with a wide range of intensity illumination. Two conditions very difficult to accept. "Direct" light has, as main addressee, the sun, the only current source of light that sends parallel rays. The problem is, now, that the quality and intensity variation of sun light can hardly be simulated by what we have as computer direct lights.

What is, to finish, a "spot" light? A spot light is a somewhat more sophisticated device, derived from the work of Warn (1983), that tried to extend the limits of local illumination models that started with Phong (1975) and used the type of lights just mentioned. It overcomes some of the drawbacks of point lights as it allows us to simulate some of the characteristic effects of real lights such as an intensity oriented distribution and the falloff at the limits between shadow and light thanks to the "cone delta angle". This attempts to approximate what happens in reality as a consequence of the fact that, as there are no "point" lights but, rather, "volume lights", we have transition regions between "umbra" and "penumbra", between parts of the surface that receive all the light available form the source, and parts that do not receive light at all. The problem is now that, in the majority of cases, light rays distribution is not constrained to conic volumes as the ones used by spot lights but, rather, to spheric volumes centered on every point on the surface of the emitting source volume. This only happens, more or less, in the case of "real spot lights" such as the ones used by photographers in their studios.

But, to avoid the excessively artificial effects that result from that, photographers use white umbrellas, shades or any other system that might allow them to minimize the excessive contrast upon the model. The only thing we have, in local reflection models, to avoid these artificial effects that can only please the computer mad fans or those people with the visual sensitivity of a bat, is "ambient" light or the recourse to other lights (point, direct or spot) to attenuate the contrast. Of course we can reach very brilliant results in this way. But they use to please some kind of observers; others, whose opinion we value more, are very conscious of the superimposed tricks and the unnatural aspect of the scene.

Is there anything better on the horizon? 1 think that the answer cannot come other way round that from simulation, less abstract and more real, of physical and geometric characteristics of source lights. This means that we will not get convincing light effects until there will be a way of treating source lights as objects that create umbra and penumbra regions and modulated intensity fields around them. This leads us, to start with, to the difference between local and global models that will be commented on the 4 and 5 point of this paper. Some recent works, as in Campbell (1990) or Picott (1992), that use lineal and area lights, may provide an answer in a near future.

If we forget about formal factors and about relative position of objects and lights, the color of a surface is sufficiently defined by a function that specifies, for each point on the surface, its reflection percentage, in a band relatively narrow of the visible spectrum. A band that is given typically a value of 10 nanometres (about 40 samples) but that can be wider (about 12, 8, or the usual 3 samples). The color of a surface will be obtained in association with a spectrophotometric curve that will gives us something like the radiography of its chromatic behavior. This is what we call the 1ocal color" of a surface considered as an optical surface, that is, with no other physical consideration in mind.

If we do not forget about formal factors and objects and lights relative positions, the color of a surface is defined by its local color plus a variety of parameters that collect the intensity and color variations due to: 1) the deviation of the surface with respect to an optimal orientation that will provide full illumination. This is usually expressed by two vectors: a unitary vector normal to the surface and a unitary vector coincident with the direction of light. When the angle between these two vectors is V intensity is maximum, when it is 900 intensity is null; intermediate values can be computed between these two extreme values. 2) the lack of illumination on parts of the surface that remain hidden from light, due to the micro-geometry of the surface itself. This effect, viewed from some distance, is not perceived as local shading or shadow but as a modification of surface color.

We use the ambiguous term "texture" to design a general effect that depends on the combination of these factors. Texture can be produced by modifications of local color, strictly speaking, as it happens, for instance, in a plain surface like polished marble, that presents surface patterns that can be described in terms of local color changes. Texture can also he produced by what we call "roughness" which is nothing else as the effect produced by local micro-forms and their shades and shadows, and which might be described exactly as we would describe the effect of a landscape or a city viewed from the sky, as it happens, for instance, in a sandstone surface of uniform tone. These materials show a characteristic texture, precisely in as much as the uniformity of their local color allows us to imagine the characteristics of their surface form.

The only correct way to model a material texture would be to model **both** its local color and its surface micro-form. Obviously, this implies an unacceptable computing cost. But it constitutes the most accurate reference to judge the clever but insufficient recourses that we have at the present moment. As it happens with the development of these kind of techniques, smart and sophisticated as they are, this reference tends to remain progressively forgotten by the complexity and variety of technical tricks and by the rapidity with which we get used to them.

Resuming many different kind of works, we can say that material and surface simulations, at the moment, is based in four main techniques:

1) The projection of images, captured from the real world, upon geometric surfaces (texture mapping). This started with the works of Catmull (1974) and Blinn (1976, 1978) and has been developed thereafter by many studies, among them those by Feibush, Cook, et al. (1980), Crow (1984), Glassner (1986) or Heckbert (1986) who provides a good summary of ten years development in this field.

2) The generation of color patterns upon geometric surfaces by means of mathematical functions (procedural modelling). This also began with the works of Catmull and Blinn and has been developed thereafter by, among others, Perlin (1985) and Peachey (1985). Functions can be unidimensional, bidimensional or three dimensional. The first are rare. The second are quite frequent though may encounter some difficulties in certain cases due to the geometry of the form. The third ones provide efficient models although the result are too often unexpected and difficult to control. It stands as one of the most promisory areas of research as it provides the most compact and portable way of specifying material attributes.

3) The apparent modification of the geometry of the surface itself by means of alteration of the surface normal during rendering (bump mapping). This was initiated with the works of Blinn (1978). It has the inconvenience that objects external contours remain the same and this results in a visual inconsistency that, in too many cases spoils the effect. It does not either, deal adequately with the interplay of light and surface material.

4) The modification of the surface geometry by means of local geometric displacements during

rendering (displacement shaders). This started with the works of Csuri (1979) by algorithmic expansion of the data describing the shape of the object and has been continued by a variety of works among them those by Fourier et al. (1982) and Harioyama et al. (1984) subdividing stochastically point on the surface to produce and displace other vertices; those by Hiller (1987) displacing directly points of the surface taken as input texture values during a subdivision algorithm; those by Cook (1984, 1987) dividing the surface in micropolygons, in the range of a sub-pixel size, also displaced by texture values. All these techniques are still been developed but it seems they not gather adequately either the interplay of light and surface. If this is so, they remain almost inefficient for quite a large variety of architectural materials.

We should add to all this, the simulation of surface materials by algorithms that take into consideration the physical properties of the materials that constitute them and that allow the renderer to simulate the local reflection material behavior by means of a function that specifies its bidirectional reflection distribution (BRDF). These was initiated with the works of Cook and Torrance (1982) and seems to be other promising way of research that should be followed by those interested in the simulation of the interplay of light and architectural materials.

4. Local Illumination Models

The first illumination model that encountered a wide success and made people talk about the "amazing photorealism" of computer graphics, was due to Buoi T. Phong (1975). Phong model was preceded mainly by those propounded by Wylley et al. (1967) and Gouraud (1971). None of these gave the least suggestion of a light presented in the scene. Phong model, on the contrary, for the first time, showed reflections that, besides, could he manipulated, making them concentrate or spread about the surface. This, was expected, should be translated in an impression related to the perception of a different type of surface. Phong model is still widely used but, once passed its well deserved success, its limitations become more and more evident. First, the created "impression" is ambiguous; an spread reflection can be due as much to the nature of the surface itself as to the nature of the light itself; a variety of observers have perceived the chan e of highlights size, in Phong famous test spheres, as much as a difference in the kind of material, as a difference in the size of the light or in the distance between this light and the material. This second interpretation, however, tends to be favoured, due to the "plastic" aspect of surfaces illuminated with Phong model. This is the second notorious limitation of this model.

There have been various works carried along with the explicit intention of overcoming these limitations. The better knowns are those by Cook and Torrance already mentioned. These limitations, however, are due to the nature of the model itself. A local reflection model ignores necessarily the interaction between the various objects in the scene. Each object, as a Leibniz monad, remains closed in its own world, illuminated by its own light of which nothing is shared with the closest neighbor it may encounter. The punishment carried by this selfish attitude is an inevitable accumulation of lights that have to multiply around to simulate an interaction that does not really exist. Anyone that have worked some time with this kind of models knows that one is forced to develop a collection of artificial tricks to give the scene an aspect of normality that is never satisfactorily reached.

5. Global Illumination Models

Global illumination models are currently classified in two groups. Those dependent of the view point, like ray tracing, and those independent of the view point, like radiosity. This division follows naturally from a well known phenomenon: shadows remain steady but reflections do not. They move if we move, and there is not a fixed position for them on the surface of the object on which they appear. Any rendering based on specularity should be, for this reason, dependent on the view point.

Ray tracing starts with a very simple notion but one that could not be developed until there was enough computer power. From a virtual view point and for every pixel on a virtual screen, a ray is sent that will intersect any surface it finds on its way and will change its trajectory depending on the geometric and optical characteristics of this surface. The ray trip will finish on a light source, at the void or at the moment it finds a surface which is specified as too deep for the number of deviations allowed by the user.

The works of Kay (1979), Greenberg (1979) and Whittted (1980) gave the first basis to a system that it can be implemented easily on a PC, provided that we do not use a too complex 3D model and that we have enough patience to wait for a couple of hours before trying still another change of parameters. There are also few programs that can translate easily 3D models generated with AutoCad to a data base composed of polygon meshes and quadrics that can be edited and adjusted with a very elementary knowledge of programming formalities.

The results can be quite impressive for some people and are almost acceptable for others in the case, mainly of models consisting of specular surfaces. The problem is, first, that shiny surfaces are rather an exception than the rule in the environments we use to live. And, second, that shiny surface do not use to behave like clean or dusty mirrors. Images generated by ray tracing show inverted images, let us call them "daughter images" that start right from the feet of their, let us call them, "mother images". But this only happens in the case of mirror or very glossy surfaces or if we look at a surface at a very low angle from its tangent plane. Models generated by ray tracing belong to an ideal world where real materials cannot find their place. But, as this world is presented in a very convincing way, the result is not shortness of representation but, quite plainly, pure deformation. Most architects would prefer a flat shading representation of their models, that do not try to be "realistic" but only clarify some aspects of form.

Radiosity starts from a concept which is also rather simple in itself but that becomes rather complex in its application, development and optimization. Every object, as it reflects some part of the light it receives, emits light, that is, it can be considered as a source of light by itself. And so, every object, in a particular environment is, on the one hand, receiving light from as many objects as it has around it in the scene under consideration and, on the other hand, sending light to all objects that surround it. Therefore, "all we have to do", is to compute the results of this interaction to obtain the value of intensity and color of every point on the surface of every object. This amounts to nothing else than solving, for a scene modeled with, let us say 1000 patches, 1000 equations with 1000 unknown Bi values (radiosity equations).

The works of Goral, Torrance, Greenberg et al. (1984) and Nishita and Nakamae (1985) gave the first basis to a method that has recently started to be applied on some PC with a very simplified basis. The authors of the first systems have very recently, Greenberg et al. (1992) themselves criticized some of the results obtained and have proposed new techniques to overcome some of the defaults of their system. There is, to start with, a too great distance with view dependent methods: we can complain about the fact that ray tracing shows us a "too specular world". But we do have, quite often, some reflections around. And these are, obviously, view dependent. Among other things, radiosity is forced to be intermingled with other methods to provide for this. This has been already tried in many works produced between 1987 and 1993, as in Wallace, Cohen and Greenberg (1987) but results do not seem to be convincing.

As it happens with modelling, perhaps the answer will come from a variety of techniques that can be combined together depending on the case we are dealing with. A unique method is something that, for methodological reasons, should be looked after. but, for practical reasons, technological history proves us that eclecticism is the most common answer to get good results.

6. Modelling for Rendering

The characteristics of the illumination models we use are related to the characteristics of the geometric models. But there we have, also, quite a variety of techniques at our disposal. From this comes a question that seems to be purely technical but that points to a more general context. Every professor involved in CAAD teaching has listen to this question: which is the best way to model this particular wall with this particular window and this particular profile? The only answer to such a question, as we all know, is "it depends" Of what? Of the modeler's intentions and the model destination.

A simple rectangular wall, with a simple rectangular window in the middle, can be modeled in AutoCad, as far as I know, in 7 different ways. Ordered from less to higher memory cost, these ways are : 1) using the command "trace" (nostalgics) or "pline", setting the width line to the width of the wall, and extruding the surface so created to equal the height of the wall and the lower and upper parts of the window; 2) using the command "3Dmesh" to create directly a polygon mesh (this is better done with an AutoLisp routine to easy the input); 3) using the command "ai_box" to use 4 "box" primitives; 4) using the command "3DFace" to create directly polygons with invisible edges so that we can use it directly for a line plot, something that cannot be done in the previous cases; 5) using the command "Pface" in a similar way; 6) using the commands "solbox" (AME) and "solsub" to create two solid primitives and substracting one from the other. 7) using the commands "pline" "solext" (AME) and "solsub" id.) to create to solid extruded polylines and substracting one from the other. The last two produce a file four times bigger than the previous ones.

Which of these options should we choose? If the model destination is a line drawing and we want to save memory the best option is surely 4 or 5. If the model destination is a simple rendering we can use, safely, options 1 or 3 which are easier to model. If the model destination is a complex rendering in which we intend to modify some aspects of shape, the best option is probably 2 which can be translated by some renderers to bicubic patches and provides a more compact and easy to interpolate representation. If the model destination is a project and we want to modify the wall or the window after they are generated, the best option is probably 6 or 7. If the model destination is another program in which we need to use the model as the point of departure for new data that shall be used for, let us say, a constructive or structural analysis then the best options are probably 6 or 7.

This is what results from such a simple example as a wall and a window, and a relatively simple modelling program as AutoCad/AME. The diversity of options grows if we consider more complex cases and we extend the variety of available techniques and programs. Some of these options should be taken into account when a renderer program is to he developed.

A sphere generated with AutoCad with a mesh density of 16 becomes a polygon mesh that occupies 50 pages of code. The same sphere, defined as a quadric, that is, in way perfectly adequate for a ray tracer that can use this definition to calculate directly the point of intersection between the ray and the surface, occupies exactly a line of code. We have saved 99% of memory and we have reduced processing time from 2 hours (for a scene including the sphere and a few more simple objects) to 6 minutes.

Are these purely technical questions or do they have something to do with architecture? Strictly speaking they are mere technical questions that we should not care much about. Nevertheless, they appear at the limits of an attitude that does have to do with architecture, that reaches, in fact, an essential point: the *intentions* that drive the generation of a model. Managing objects or managing appearances is an essential choice that has to be accepted, from the point of view of architecture, by both ends. To reject this duplicity of paths to follow, means to reject architecture in itself.

7. Interfaces. The RenderMan Interface

The word "rendering" is well established in english literature on visual arts. Perhaps it was used for the first time, in the computer graphics world at 1981, a year in which L. Carpenter developed for Lucas Film what it is sometimes considered as the first or one of the first rendering systems, the REYES ("Render Everything You Ever Saw") that was used to create special effects for the film "Startrek II". Some very well known people in this world, like R. Cook and E. Catmull, participated in later developments when different techniques and platforms were coming together and was growing the necessity of having some common type of protocolo that would allow the portability of results.

This need to count with a common language that could allow programmers and users to describe a scene in a generic way that could be shared by different people was the point of departure of the RenderMan interface language that was developed mainly by P. Hanrahan in a first version that circulated widely at the lates 1980 and in which he was joined by B. Reeves and R. Cook, among others. In words of Hanrahan himself : The RenderMan interface is meant to be the Postscript of 3-D graphics. Just as a Postscript allows a desktop publishing system to pass page representations to a printer, RenderMan allows three-dimensional modeling systems to pass scene descriptions to a renderer. " This account emphasizes a fundamental characteristic: this interface language wants to stay between the modeler and the renderer. It is, precisely, "a methodology of scenes description", meaning by "scenes" a collection of objects, cameras and lights.

In contrast to a modeling system, a rendering system is relatively steady. Objects, in general, exhibit a formal diversity that is impossible to hold together by a unique strategy. We should accept the idea that, probably, there will he as many modeling systems as attitudes and intentions in design. On the contrary, interaction of color and light versus objects, despite the great complexity of the phenomena that they imply, and that, for the moment, are still far away from what can be simulated by available techniques, can be embraced by a few number of related laws that serve as a basis to reproduce the main effects we wish to evoke. Besides, this can be gradually approximated, step by step, taking some profit from the previous one.

This proposal of standardization presents some characteristics that it is worth to underline. First, it focuses on the appearance of objects; it considers shade, not shape. This implies a crucial difference between rendering and modeling. Rendering becomes a description language that manages only *shaders* classified in three main groups: light shaders, surface shaders and atmospheric shaders. These shaders constitute the main body of the language and can accept every known model: any type of light or illumination model is acceptable as a generic description of a "light shader", from Phong to Ray tracing or Radiosity or linear and area light sources; any surface definition can become a "surface shader", from a map projection that starts with a photographic image till a prcedural generation based on fractal subdivision, local color specification based on an arbitrary number of samples, bump mapping, or displacement shaders. This may overcome actual limitations and leave the way open for new developments.

On the modeling side, it favors the conversion to rendering generic primitives that consist mainly on bicubic patches that allow a more compact representation and more interpolation capacity. It also includes, a s primitives, polygonal meshes, bilinear patches or nurbs (non-rational-b-splines). The conversion depends on the modeling program; this means that any strategy is acceptable as the renderer does not interfere with the way the data is generated and the type of objects produced. This maintains the independence of intentions that otherwise might collide. The converted files in RIB (RenderMan Interface ByteStream) can be edited; this allows some further degree of optimization as in the example above when an sphere generated in AutoCad as a huge polygon mesh could substituted by a simple quadric.

A growing number of software producers seem to be using this interface as a frame for their developments. Autodesk has also included a render module with AutoCad that uses not only the language but the some of the original examples made by the developers at Pixar. It has a few limitations and errors that have prevented it to be more widely used. But apart from what will the future bring out of this proposal, the main question is to understand that such a common interface or any other that may substitute it is, among other things, the way to better reveal the actual limitations and the way to overcome them.

8. What should we expect?. The Architect point of view.

Abstraction, from the point of view of an architect, has a literal meaning which should have priority in relation with the common meaning given to this word by journalists associated to the art of this century. It means, before anything else, an intellectual operation that help us to select some characteristic features of an object, and to generalize or universalize them, that is, to put them in a meaningful relationship with similar features of other objects. To design is to abstract: it means establishing a relationship between some particular features of a problem (the program we have to conform) and some particular potentiality inherent to some forms or some formal structures.

An architect should always prefer a simple abstraction, very general, but appropriate to the model, to an artificial presentation, that seems to be very detailed but misrepresents it. In other words: we should prefer a representation that does nothing but indicate, with no detail at all, the chromatic profile of a shape (in such a general way as to suggest whether it has a "warm" or "cool" tendency) and the main lines of its internal contours, by means of a technique so crude as "flat shading" to a representation that distorts the form with violent shadows or artificial textures that try to simulate, without success, some particular material. The "flat shading" is not trying to be realistic; so we accept willfully those indications that do not pretend passing beyond what they can inform about form attributes. The artful simulation becomes irritating, because it seems to be taking for granted that our visual sensitivity is so coarse that we will accept easily a rough simulation.

From the point of view of an architect that tries to keep alive some interest on the nature of materials and their interaction with form, renderings are a line of research and development that still has a long way to follow. Some of this work might be done through the generation of materials libraries that could be shared among different people at different places. Other, like types of lights or global illumination models, depend upon complex technical developments that very few architects can follow and collaborate with. Concerning the first point there are two main questions that should he posed :

1. We need an extensive research on procedural methods that, should they prove effective, can replace with notorious advantages projective methods, like texture mapping.

2. We need an interface protocol that can guarantee the portability of interchanges. Perhaps RenderMan is an answer to this question but this has to he confirmed by general practice.

9. What should we expect? The School point of view.

Abstraction, from the point of view of an student, means the confrontation with an universe of powerful relationships, whose strength is felt and respected but that appears as a labyrinth without keys or tracks that help to go across it. That is the reason why a student, characteristically, has a tendency to use pedantic terms when speaking, while grasping very definite examples, models or images when designing. Designing, for an student, implies, very often, trying to impress, to dazzle, to wave flags around, throwing images that might serve him or her as a shell. This description does not pretend to be caustic; that is the way we all have followed to reach abstraction. There is not any other. But, just because of that, the purpose of the school should be a double one. On the one hand, it must open and keep in front of students, the treasures chest, where all images of our cultural tradition are kept. On the other hand, it must submit all these images to a hard critic.

The rendering programs we have at the present moment at our disposal, well known programs like 3DSttidio, Autodesk RenderMan, various Ray tracing programs, etc., or any other which has succeeded to implement a more or less friendly version of the scarcely half dozen "light shaders" that have appear between 1973 and 1993, due to Gouraud, Phong, Whitted, Greenberg .... among others, are a double-edged weapon. They can help some students to express some ideas in an effective way. But they can also provide a way to misunderstanding that makes him myopic to the real behavior of materials and to the subtle interplay of light and forms. There are many examples that prove that this is, unfortunately, the case. From the point of view of the school, a critical assimilation of this process is required, and one that might help to determine exactly their distortions and to indicate the correct way of development.

1. MONEDERO, J., REDONDO, E. "Electronic Collage". Barcelona, ECAADE 92 Proc., pp. 241-250.
2. MONEDERO, J., CASAUS, A. "Structured data versus open data". Barcelona, ARECDAO 93, pp. 353-368.

References

1967 WYLLIE, C:, ROMNEY, G.W., EVANS, D.C., ERDHAL, A.C. "Halftone perspective Drawings by Computer" *Proc. Fall Joint Computer Conf.* Thompson Books, Washington D. C.

1968 APPEL, A. "Some techniques for machine rendering of solids". *AFIPS Conference Proc.* 32, 37-45.

1969 WARNOCK, J. *A Hidden Surftice Algorithm for Computer Generated Halftone Pictures.* Computer Science Dept.C.S. Tech. Report 445, University of Utah.

1971 GOURAUD, H. "Computer Shading of Curved Surfaces". Ph. D. dissertations, University of Utah. Also in: *IEEE Transaction on Computers* Vol. TC-20, June 1971.

1974 CATMULL, E. 1974 "A Subdivision algorithm for the display of curved surfaces. " Ph.D., University of Utah.

1974 SUTHERLAND, I.E.; SPROULL, R.F.; SCHUMACKER,R.A. "A Characterization of Ten Hidden Surface Algorithms". *ACM Computing Surveys,* 6(1), March 1974, 1-55.

1975 BUI-TUONG PHONG "Illumination for computer-generated pictures". *Comm. CM*, 18 (6), 311-17.

1975 BUI-TUONG PHONG & CROW, F.C. "Improved rendition of polygonal models of curved surfaces". *Proc. 2nd USA-Japan Computer Conf.,* pp. 475-80.

1975 CATMULL, E. "Computer display of curved surfaces". *Proc. IEEE Conf. on Computer Graphics, Pattern Recognition and Data Structures.* May, 1975 Reprinted in : Freeman H. ed (1980). *Tutorial and Selected Readings in Interactive Computer Graphics*, New York (IEEE) pp. 309-15).

1976 BLINN, J.F. & NEWELL, M.E. 7exture and Reflection in Computer Generated Images". *Communications of the ACM* 19 (10) 542-547.

1977 BLINN, J. F. "Models of light reflection for computer synthesised pictures". *Computer Graphics*, 11 (2), 192-8.

1978 BLINN, J.F. -Simulation of Wrinkled Surfaces". *Computer Graphics* 12 (3) 286-292.

1979 CSURI, C. et al. "Towards an interactive high visual complexity animation system*". Computer Graphics (Proc. SIGGRAPH 79)* , Vol. 13, pp. 289-299, 1979.

1979 KAY, D.S., GREENBERG, D.P. "Transparency for computer synthesised objects*". Computer Graphics*, 13 (2), 158-64.

1979 KAY, D.S. "Transparency, refraction and ray tracing for computer synthesised images". *Masters Thesis.* Cornell University. 1980 FEIBUSH, E.A., LEVOY, M., COOK, R.L. "Synthetic Texturing Using Digital Filters". *Computer Graphics* Vol. 14, No. 3, July 1980, pp. 294-301.

1980 WHITTED, T. "An Improved Illumination Model for Shaded Display". *Communications of the ACM* 23: 343-349.

1981 KAJIYA, J.T. "Ray tracing parametric patches". *Computer Graphics,* 16, (3), 245-54.

1982 COOK, R.L. & TORRANCE, K.E. "A Reflectance Model for Computer Graphics". *Computer Graphics* 15: 3 307-316.
1982 FOURNIER, A., FUSSELL, D., CARPENTER, R.L. "Computer rendering of stochastic models". *CACM*, Vol.25, pp. 371-384.
1983 KAJIYA, J.T. "New techniques for ray tracing procedurally defined objects". *Computer Graphics*, 17 (3), 91-102.
1983 WARN, D.R. "Lighting Controls for Synthetic Images". *Computer Graphics*. Vol. 17, No.3, pp. 13-21.
1984 BARR, A. H. "Decal Projections". *Computer Graphics (Proc. Siggraph 84)*, Course no. 15, july 1984.
1984 COOK, R.L., PORTER, T., CARPENTER, L. "Distributed Ray Tracing". *Computer Graphics (Proc. SIGGRAPH 84)*.
1984 COOK, R. L. "Shade Trees" *Computer Graphics*, Vol. 18, No.3, pp.223-23 1.
1984 CROW, F.C. "Summed-area tables for texture mapping". *Computer Graphics*, 18 (3), 207-12.
1984 GORAL, C. M., TORRANCE, K. E., GREENBERG, D. P., BATTAILE, B. "Modeling the Interacion of Light Between Diffuse Surfaces". *Computer Graphics* (Proc.Siggraph), Vol. 18, No. 3, July 1984, pp. 213-222.
1984 HECKBERT, P.S. & HANRAHAN, P. "Beam tracing Polygonal Objects". *Computer Graphics* 18, (3), 119-145
1984 HECKBERT, P.S. "Survey of Texture Mapping". *IEEE Computer Graphics & Applications*, Vol.6, No. 11, Nov. 1984, pp. 56-67.
1984 VERBECK, C.P., GREENBERG, D.P. "A comprehensive light source description for Computer Graphics". *IEEE Computer Graphics & Applications*, Vol.4, No.7, pp.66-75.
1985 COHEN, M.F. & GREENBERG, D.P. "The Hemi-Cube. A Radiosity Solution for Complex Environments". *SIGGRAPH 85*, 31.
1985 GAGALOWICZ, A., MA, S.D. "Model driven synthesis of natural textures for 3-D scenes". *Proceedings Eutographics 85*, pp. 91-108, sep. 1985.
1985 NISHITA, T., NAKAMAE, E. "Continuous Tone Representation of three Dimensional Objects taking account of Shadows and Interreflection". *Computer Graphics (Proc. SIGGRAPH 85)*, Vol 19, No.3, 23-30.
1985 PEACHEY, D.R. "Solid Texturing of complex surfaces". *Computer Graphics (Proc. SIGGRAPH 85)*, Vol.20, No.4, 55-64.
1985 REEVES, W.T., SALESIN, D.H. "Rendering Antialiased Shadows with Depth Maps". Computer Graphics, Vol.21, No.4, pp.283-291.
1986 COHEN, M.F., GREENBERG, D.F. & IMMEL, D. S. "An efficient radiosity approach for realistic image synthesis". *IEEE Computer Graphics and Applications*, Vol.6, No.2, pp.26-35.
1986 GLASSNER, A.S. "Adaptive precision in texture mapping". *Computer Graphics (proc. SIGGRAPH 86)*, vol.20, pp.297-306.
1986 GREEN, N. "Environment Mapping and Other Applications of World Projections". *IEEE Computer Graphics and Applications*, Vol.6, No. 11, Nov. 1986, pp. 108-114.
1986 HECKBERT, P.S. "Survey of texture mapping". *IEEE Computer Graphics and App.*, Vol.6, pp.56-67, 1986.
1986 IMMEL, D.S., COHEN, M.F., GREENBERG, D.P. "A Radiosity Method for Non-Diffuse Environments". *Computer Graphics* (Proc. Siggraph), Vol.20, NO. 4, Aug. 1986, pp 133-142.
1986 KAJIYA, J.T. & KAY, T.L. "Ray tracing complex scenes". *Computer Graphics*, 20 (4), 269-78.
1986 KAJIYA, J.T. "The rendering equation". *Computer Graphics*, 20 (4), 143-50.
1986 YAEGER, L., UPSON, C., MYERS, R. "Combining physical and visual simulation-Creation of the planet Jupiter for the film '2010' ". Computer Graphics, Vol.20, pp. 85-93.
1987 CARPENTER, L., CATMULL, E. & COOK, R.L. "The REYES image rendering architecture". *Computer Graphics*, 21 (4), 95-102.
1987 NISHITA, T. "A shading model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources". *Computer Graphics (Proc. SIGGRAPH 87)*, Vol.21,No.4, pp.303-10.
1987 WALLACE, J.R.; COHEN, M.F.; GREENBERG, D.P. "A two-pass solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods". *SIGGRAPH 87*, 331-320.
1988 COHEN, M.F., CHEN, S.E., GREENBERG, D.P.& WALLACE, J.R. "A progressive refinement approach to fast radiosity image generation". *Computer Graphics*, 22 (4), 75-84.
1989 SILLION, FX, PUECH, C. "A General Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods". *Computer Graphics (Proc. Siggraph 89)*, Vol.23, No.3, July

1989, pp. 335-344.
1990 CAMPBELL III, A.T., FUSSELL, D.S. "Adaptive Mesh Generation for Global Diffuse Illumination". *Computer Graphics* (Proc. SIGGRAPH 90) Vol. 24, No. 4, Aug. 1990, pp. 155-164.
1990 UPSTILL, S. *The RenderMan Companion.* Reading, Mass., Addison-Wesley.
1991 HANRAHAN, P., SALZMAN, D, AUPPERLE, L. "A Rapid Hierarchical Radiosity Algorithm". *Computer Graphics* (Proc. Siggraph), Vol.25, No.4, July 1991, pp. 197-206.
1991 SILLION, F.X., et al. "A Global Illumination Solution for General Reflectance Distributions". *Computer Graphics* (Proc. Siggraph), Vol.25, No.4, July 1991, pp. 187-196.
1992 HECKBERT, P.S. "Discontinuity Meshing for Radiosity". *Eurographics Workshop on Rendering.* May 1992, pp. 203-216.
1992 LISCHINSKI, D. TAMPIERI, F. GREENBERG, D.P. "Discontinuity Meshing for Accurate Radiosity". *IEEE Computer Graphics & Applications*, November 1992, pp.25-38.
1992 PICOTT, K.P. "Extensions of the Linear and Area Lighting Models". *IEEE Computer Graphics and App.,* March 1992, pp. 31-38.

Order a complete set of
eCAADe Proceedings (1983 - 2000)
on CD-Rom!

Further information:
http://www.ecaade.org