

A SEMI-AUTOMATED CLASSIFICATION METHOD FOR INFRARED SIMULATION

OECD CONFERENCE CENTER, PARIS, FRANCE / 28–30 JANUARY 2014

Carole Nissoux ⁽¹⁾, Yann Dupuy ⁽¹⁾, Carlos Marmelo ⁽¹⁾, Thierry Cathala ⁽¹⁾

⁽¹⁾OKTAL Synthetic Environment, 11 Avenue du Lac, 31320 Vigoulet-Auzil (France)

Email: carole.nissoux@oktal-se.fr

KEYWORDS: simulation, scene and target generation, scenario

ABSTRACT:

As years went by, virtual environments have become much more complex. The computation and rendering growing capacities allowed synthetic environment users to have much bigger mock-ups (several thousands of square kilometres), with much more detail, much more precision (up to 50 cm precision on aerial photos). When dealing with infrared rendering, this means having a lot of images to classify (i.e. associating to each pixel of a texture a physical material suited for infrared rendering), especially aerial photos that are the basis of the mock-ups terrain. Manually classifying those terrain textures is costly... and tedious. Moreover, manual classification of big sets of aerial photos, especially if several operators take part in the process, can lead to discontinuities and inconsistencies. The solution to this problem is automated, or semi-automated, infrared classification. Automated classification of a texture means being able to automatically associate to a pixel of a texture a physical material or a physical materials composition, in a more or less certain way. This paper presents some automated classification methods and why they cannot be used, in our framework, on a daily basis. It then roughly describes the semi-automated classification method created by OKTAL-SE and the spectral validation method used to validate this classification method. Comparative results, obtained by automatically classifying an existing and already manually classified virtual mock-up, are then presented. Finally this paper discusses future enhancements allowing getting round existing limitations.

1. INTRODUCTION

As years went by, virtual environments became much more complex. The computation and rendering growing capacities allowed us to have much bigger mock-ups, with much more details, much more precision.

When dealing with infrared (IR) rendering, for

OKTAL-SE this means having a lot of images to classify, especially aerial photos that are the basis of the mock-ups terrain.

Manually classifying those terrain textures is costly... and tedious. Moreover, manual classification of big sets of aerial photos, especially if several operators take part in the process, can lead to discontinuities and inconsistencies.

The solution to this problem is the automated, or semi-automated, IR classification.

Automated classification of a texture means being able to automatically associate to a pixel of this texture a material or a materials composition, in a more or less certain way.

This document will study several automated classification methods, detail the method chosen by OKTAL-SE, present a validation method and discuss future enhancements.

2. AUTOMATED CLASSIFICATION METHODS FOR IR DATABASE

When dealing with IR automated classification, it is quite clear that having multispectral images can grandly simplify the problem.

If we have an aerial photo of a field of green grass with, in the middle, a house with a concrete roof, painted the same green as the grass. Using only the RGB visible information would lead to associate the same material to the pixels representing the field and the roof though their radiative and thermal properties are very different. The same problem occurs when dealing with pixels representing elements with the same material but one being in the shadow and the other in the sun.

When having multispectral images, such as Landsat photos, the automated classification becomes much "easier".

For example we could use the **retro-computation method**. This method consists in analysing every pixel of the texture to classify, in all the available spectral images and deduce its physical spectral

properties forming the “real” material that this pixel represents. Once it is done, for each pixel, we browse the SDM material library to find the “closest” material to this real material.

The metric used to determine the distance between two materials can be, for example, based on the difference between all the properties values. Those value differences could be weighted according to the importance we give to some of the properties, to the degree of certainty we have on how those properties were deduced.

But, as we can imagine, deducing the physical spectral properties of a material from several radiance information is not that easy. Especially since the spectral response of a terrain is completely dependent on the atmospheric conditions when the aerial photo was taken, the date, the hour, etc., information that we do not precisely have. It is the same for the gain applied to the photo that can drastically modify the information we perceive.

The conclusion is that this method could be very precise... but requires information we cannot have for sure.

The **scattergram method** is somehow simpler to use. In this case, we do not try to deduce the physical spectral properties of each pixel but we try to take into account the ratios between the different pixel values in the different domains. A scattergram is a hypercube where we can materialise those ratios.

In a scattergram, if we affect an axis for each available spectral photo, we can locate each pixel of the texture by giving it, on each axis, its value as coordinate (RGB colour for visible image, radiance for IR images). If the number of dimensions of the scattergram is sufficient and the spectral images are in domains that are differential enough, it is clear that two pixels that are close in the scattergram share the same material.

Once it is done, the only problem remaining is to determine which material of the SE-WORKBENCH material database is shared by those pixels, or at least which is the closest. This could be done by spotting the different SE-WORKBENCH materials in the scattergram and affecting a SE-WORKBENCH material to the closest pixels in the scattergram, or by choosing a minimal and maximal allowed deviation and affecting a SE-WORKBENCH material to all the pixels that are in the scattergram areas corresponding to this material.

But this problem is not so simple. It supposes to build a test database with all the SE-WORKBENCH materials and generate multi-spectral images similar to the images used for the

classification (same atmospheric condition, same gain, etc.) in order to be able to locate our materials in the scattergram.

So even if this method is simpler than the retro-computation method, we find the same issue: it requires information we cannot have for sure.

Moreover, those two methods, as well as other methods that could be based on multi-spectral images... presume that we have multi-spectral images. But it is almost never the case. The only “usual” case where we have multi-spectral images is when we use Landsat data. But Landsat images have a 16m precision, which is not always enough for the virtual environments we build, and their spectra are not differential enough to have a high quality result with those methods that remain costly to apply.

The only sure thing when using aerial photos for generating 3D database is: we always have the visible images. So even if trying to automatically classify a texture for IR from a visible image is neither simple nor sure, it is the only direction we can take if we want to use automated classification on a daily basis.

3. OKTAL-SE AUTOMATED CLASSIFICATION METHOD

3.1. Overview of the method: keep it simple

Automated classification using visible images is quite simple because we do not have too much available information: being given an aerial photo to classify, being given a SE-WORKBENCH materials set, we choose a colour for each material (that will be the reference colour/material palette) and affect this material to the pixel that are the closest to its colour in the colour cube.

But we can be a little subtler than that.

First, with a little pre-processing, we can reduce the material reference palette. We do not have to arbitrarily choose a colour for each SE-WORKBENCH material and then apply this palette to the aerial photo (that would almost be the same as choosing them at random). We can use the photo itself to give us an idea of what the “main photo colours”, the most relevant colours, are by reducing the number of colours and so choose relevant materials.

If we have several photos of a zone to classify, the only difference is that we should first assemble them (some tools do it correctly) in order to apply the process of colour reduction and choice on the whole zone, guaranteeing some continuity in the classification of each photo.

Once it is done, with an image editing software, we

can reduce the number of the colours of the image to N, N being classically 16 or 32 colours. The only work to do is then to look at those N relevant colours and decide which SE-WORKBENCH logical material corresponds (by looking at the photo or by finding information in other sources such as land cover or even Google Earth).

Once a relevant material/colour palette has been created, we can get back to our original texture(s) and start the classification process. For each pixel in the image, we find the material with the closest colour and affect it to the corresponding pixel in a classified texture.

Of course it will not solve the green grass/green concrete problem (see chapter 2). But the effects of this problem could be attenuated by performing the closest material search in the HSV (hue, saturation, value) colour frame and not in the RGB colour frame.

For example, if we have grass in the sun and grass in the shade, their RGB colours will be quite different. But in the HSV colour frame, since they share the same hue and probably the same saturation, being only differentiated by the value (or brightness), chances are that they will be considered close enough to share the same material.

We can also choose a greater amount of relevant colours that will allow us to have grass in the shade and grass in the sun colours and affect the same material "grass".

Finally, besides the green grass/green concrete problem, we also have a problem with the water. Seen from the air, rivers, lakes, ponds, are not always "as blue as the sea". They often have a greenish hue. And so it is very probable that the automatic classification method will mistake it for grass or field. The only solution is then to perform a pre-processing on the texture to classify: selecting all the pixels representing water and adding a transparency component. The automatic classification tool will then know that a pixel with a transparency component should be associated to the water material.

Once this work is done, the texture to classify can be passed, as well as the colour/material reference palette, to the automatic classification tool developed by OKTAL-SE.

3.2. Validating and tuning using Landsat data

Of course an automated classification must be validated. And the validation process can also be considered as a part of the classification process by its "tuning" aspect: once the validation data have been built, we can always fine tune our material/colour material palette using the validation results.

3.2.1. Creating validation data in several spectral bands

As we said in 2, the multi-spectral images available

are usually Landsat data that are available for virtually any area in the Globe. They are not precise enough (16m) and the spectra available are not differential enough to be used for a method based on multi-spectral images, but they can be a good material for validation.

The idea is to take the automatically classified texture we want to validate and the corresponding Landsat data. From the classified image, we build a simple SE-WORKBENCH scene modelling the area represented by the original aerial photo and put it in a SE-SCENARIO scenario.

Then, for each available Landsat spectral image, we add a corresponding context, an adapted sensor and compute a SE-RAY image that we can compare to the Landsat photo.

But things are not so simple: to do that we need to recreate an environment that is consistent with the Landsat images.

3.2.2. Retrieving consistent atmospheric data and context

To be able to perform a SE-RAY simulation and compute images that can be compared to the Landsat spectral images, we need several pieces of information:

- The position of the sensor in the scene
- The sensor characteristics
- The atmospheric conditions
- The thermal information
- The gain applied to the spectral photos

The sensor position in the scene can be retrieved from the Landsat information: the sensor is on a satellite that is at an altitude of 705km and must be placed at the centre of the area.

The sensor characteristics can be retrieved from the Landsat handbook.

Some information allowing us to recreate the atmospheric condition can be retrieved quite easily in the Landsat metadata, such as the date, but some are missing, as the hour of day. And we of course know the location.

Using that, we can retrieve the thermal information needed to compute the SE-WORKBENCH thermal data. This information is the air temperature for the 24 hours on the day the photo was taken. It can be found in meteorological database that are accessible through the Internet. At this point, the only problem is that, when dealing with a 100kmx100km area, choosing the average air temperature can be difficult.

But we still need the atmospheric conditions.

Usually, the aerial photos are taken on fine weather days so any fine weather atmospheric condition of the SE-ATM-DB (SE-WORKBENCH atmospheric database) could be fine. But we can refine that by using meteorological database that will give us more precise information about the exact weather on that day.

Another problem is that, for Landsat data for example, the sensor is at an altitude of 705 kilometres. This is a problem for SE-ATMOSPHERE that has not been designed to compute such atmospheric conditions. OKTAL-SE thus had to develop a small tool allowing the user to create "satellite weather condition".

The gain applied to the image can be retrieved from the Landsat handbook that gives abacus to compute the minimal and maximal gains for every image.

3.2.3. Validation of classification

So once we retrieved the information we needed in order to have, for each available Landsat spectral image, consistent atmospheric data, consistent sensor and sensor position, consistent gain, all we have to do is to compute the corresponding SE-RAY (SE-WORKBENCH ray tracer software) images and to compare it to the Landsat images.

Note: Other data can be used to complete this validation such as looking at land cover data or looking at GoogleEarth.

3.3. Limitation

We already know the main limitation of this method: it is the difficulty to distinguish between a green pixel representing grass and a pixel with the same green representing some concrete painted green or, two pixels with grass, one in the shade and one in the sun.

But the limitations can also come from the aerial photo itself, especially for Landsat data.

For example, Landsat data sometimes have clouds in it. In this case, the automatic classification method will not detect it and affect to the cloud pixels the material with the closest colour.

There can also be colorimetric discontinuities in a set of aerial photos representing a whole area to classify, leading to classification inconsistencies.

Those problems can force us to do a lot of image edition pre-processing, in addition to the water processing, and lead to an even more imprecise result.

4. EXPERIMENTATION

4.1. Automated classification using Landsat data

This part of the experimentation consists in applying the automated classification on an area using visible Landsat images and validating this classification using the multi-spectral corresponding Landsat data. Every step of the procedure will be detailed, including the building of the validation data.

4.1.1. Procedure

▪ Classification

The first step is, of course, to retrieve the Landsat data on the area we want to model. For this experimentation we chose the Lodeve area (a small town near Montpellier in the South of France) because we also had the BD ORTHO[®] of this area (see chapter 4.2 for more detail).

We extracted from the Landsat data the area corresponding to the area covered by the BD ORTHO[®] data in order to have a smaller Landsat photo to classify (classifying the whole Landsat photo would have required to cut the photo up into smaller photos which is costly and not interesting in our context since only a small zone interests us). The second step was to pre-process all the water areas. It was done using an image-editing tool.

Then we reduced the Landsat photo colours. We reduced the colours down to 16 colours, creating a "palette" image. We also created a 32 colours palette image in order to test the influence of the number of materials on the automatic classification. Once those palette images have been generated, we used an image-editing tool to transform them into indexed colour images in order to be able to access their colour palette.

Once the two palette images were created, we retrieved the 16 and 32 relevant colours of those images in order to create the two actual material/colour palette text files.

The construction of the material/colour palette is probably the most crucial step in the automatic classification process... and the most tedious one.

Using an image-editing tool on the palette images, for each colour of the colour palette of this image we must locate the pixels of this colour and determine which SE-WORKBENCH material is the most relevant among the available materials.

The resolution of the Landsat photo data being very low, for some areas it can be necessary to

use other information sources such as Google Earth or land cover.

Once the palettes are ready, we created two material files (one for the 16 colours palette, one for the 32 colours palette) containing:

- A material referencing the RGB texture to classify
- The list of all the logical materials listed in the palette
- The multi-domains library describing those logical materials

Then all we have to do is to use the OKTAL-SE prototype tool RGB2IPC on a colour/material palette and the corresponding material file.

The result is a classified material file referencing the classified texture corresponding to the RGB texture.

The automatic classification process is now finished. We can then create the validation data.

▪ Creating the validation scenes

As we saw it in chapter 3.2, we need various data in order to be able to validate (and possibly tune) our automatic classification.

We need:

- Validation databases
- A sensor consistent with the various Landsat spectral bands
- Atmospheric conditions consistent with the various Landsat photos
- Thermal data consistent with the various Landsat photos
- The gain applied to the spectral photos

Creating the validation databases is relatively simple. We will create a simple plate which dimensions are the classified Landsat photo dimensions (we can compute it quite simply: 1 pixel corresponds to 14,25m).

In real life, when generating a terrain database, the terrain relief and the Earth roundness influence the infrared rendering. But the satellite being at an altitude of 705km and, for a whole Landsat representing one arc degree, the altitude delta induced by the Earth roundness being about 200m, we can reasonably ignore the relief and Earth roundness and represent the scene as a plate.

We will map the classified texture on this plate. The sensor will be placed at the centre of the plate at an altitude of 705km. It is a simple sensor, with no spectral response. We will create one context per Landsat band.

The atmospheric data were generated with a spectrum covering all the Landsat wavebands and

ranges and altitudes up to 300km.

The thermal data were computed from atmospheric file generated using information found in weather database accessible through the Internet.

4.1.2. Results and limitations

▪ Gains

The gains of all Landsat images can be computed from their wavebands, their metadata and from abacuses that can be found in the Landsat handbook.

The Landsat wavebands are the following:

Landsat 7	Wavelength (μm)
Band 1	0.45 - 0.52
Band 2	0.52 - 0.60
Band 3	0.63 - 0.69
Band 4	0.77 - 0.90
Band 5	1.55 - 1.75
Band 6	10.40 - 12.50
Band 7	2.09 - 2.35
Band 8	0.52 - 0.90

The Landsat images gains abacuses are the following. They give, for each image, the low and high gains. What gain an image uses can be found in the metadata.

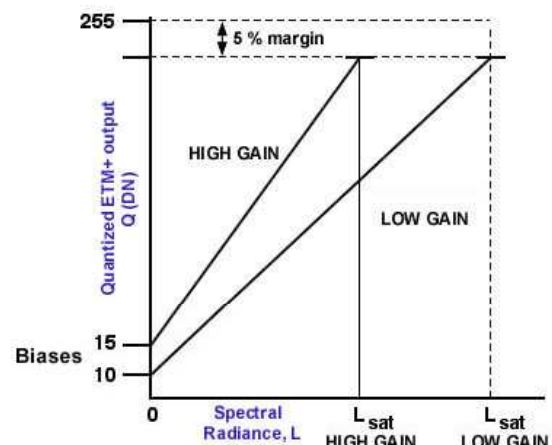


Figure 1. Landsat images gains

We first computed images in the Landsat conditions, i.e. seeing the whole area from 705km above.

The following images show, on the left, the Landsat images, on the right the images computed with SE-RAY for 16 with the Landsat gains applied for band 6 context (LWIR band).

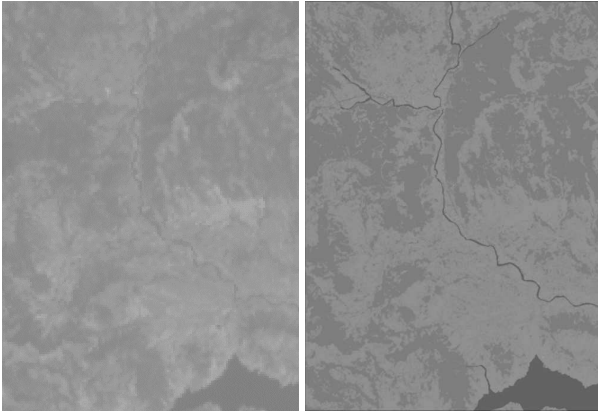


Figure 2. Band 6 reference and simulated image

So first, it seems that our atmosphere is not opaque enough: this causes the contrast to be higher than in the Landsat images. Otherwise, the classification results are quite coherent with the Landsat images.

4.2 Application to BD ORTHO® data

This part of the experimentation consists in applying the automated classification on an area using visible BD ORTHO® images and validating this classification using the multi-spectral corresponding Landsat data. Every step of the procedure will be detailed, including the building of the validation data.

4.2.1. Procedure

▪ Classifying

The first step is of course to retrieve the BD ORTHO® data on the area we want to model. As said in chapter 4.1.1, we chose the Lodeve area because we already had the BD ORTHO® data of this area.

The second step was to pre-process all the water areas. It was done using an image-editing tool.

Then we created the palettes. To cover the area we want to work on, the BD ORTHO® data requires several aerial photos. So, in order to build global colour palettes (to insure continuity in the material used), we first had to build a composite photo gathering all the BD ORTHO® photos (with a lower resolution). We work on this composite to create the 16 and 32 colours palettes.

We reduced the composite photo colour down to 16 colours and 32 colours using an image-editing tool. Once those palette images have been generated, we transform them into indexed colour images in order to be able to access their colour palette.

Once the two palette images were created, we retrieved the 16 and 32 relevant colours of those images in order to create the two actual material/colour palette text files.

Once the palettes are ready, we created the two material files (one for the 16 colours palette, one for the 32 colours palette) needed by the automatic classification tool but this time containing all the BD ORTHO® images.

Then we used the OKTAL-SE prototype tool RGB2IPC on a colour/material palette and the corresponding material file.

The result is a classified material file referencing the classified texture corresponding to the RGB texture.

▪ Creating the validation scenes

The creation of the validation scene for the BD ORTHO® data was exactly the same as for the Landsat data, described in chapter 4.1.1 (actually they were almost the same since we had to get in the same conditions) except for the database itself that was not a simple plate but a grid with the 88 BD ORTHO® textures mapped on the grid cells.

4.2.2. Results and limitations

▪ Gain

Since we wanted to have exactly the same conditions for all the images, in order to be able to compare them to the real Landsat images, the gains applied were also the same as for the Landsat simulation.

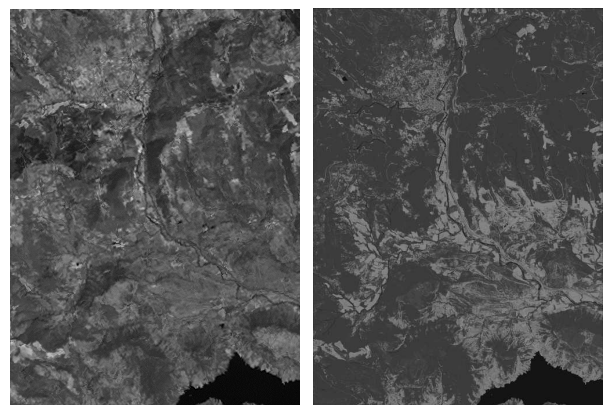


Figure 3. Band 5 reference and simulated image

So as for the Landsat simulation, it seems that our atmosphere is not opaque enough: this causes our water to be blacker than in the Landsat images and the whole image to be brighter than it should. Otherwise, the BD ORTHO® classification results are quite coherent with the Landsat images.

5. OPTIMISATION STRATEGY: USING LAND COVER INFORMATION

So as we foresaw it, one of the main limitations of our method is that, if we associate for example the ochre colour to the “wheat field” material, and we have a village with ochre roofs, the automatic classification algorithm will not be able to make the difference.

One way to get round this problem is to use land cover information.

The land cover or land use is the human use of land. Land use involves the management and modification of natural environment or wilderness into built environment such as fields, pastures, and settlements.

5.1. Method

The method itself is quite simple: instead of using a global colour/material palette on the aerial photo to classify, we use the land cover information to know, for each pixel, what is the nature of the area it is located into and perform a material search in a palette consistent with the nature of this area.

So the automatic classification method could be optimised by using, in addition to the texture to classify, a land cover file and a configuration file containing, for each type of zone of the land cover, a colour/material palette. Then for each pixel, we find the land cover zone it belongs to, retrieve the corresponding palette and perform the material search.

In practical terms the automated classification method would have an extra step (part of it could be automated too): instead of creating the global colour/material palette on the whole global texture, we would first divide it into layers corresponding to the land cover areas, then create the colour/material palette of each zone and finally start the classification process.

One must however be careful not to use too many materials (otherwise the classified texture will trespass the SE-WORKBENCH material limitations). For example, grass is grass, even if in reality there is a difference between a neat city lawn and a rough pasture grass.

5.2. Results and limitations

Some tests were carried out using a BD ORTHO[®] aerial photo and the corresponding data of the CORINE land cover.

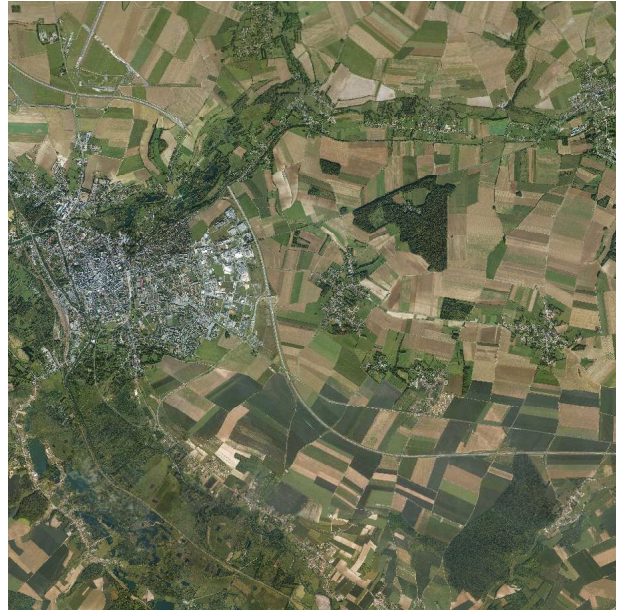


Figure 4. BD ORTHO[®] aerial photo

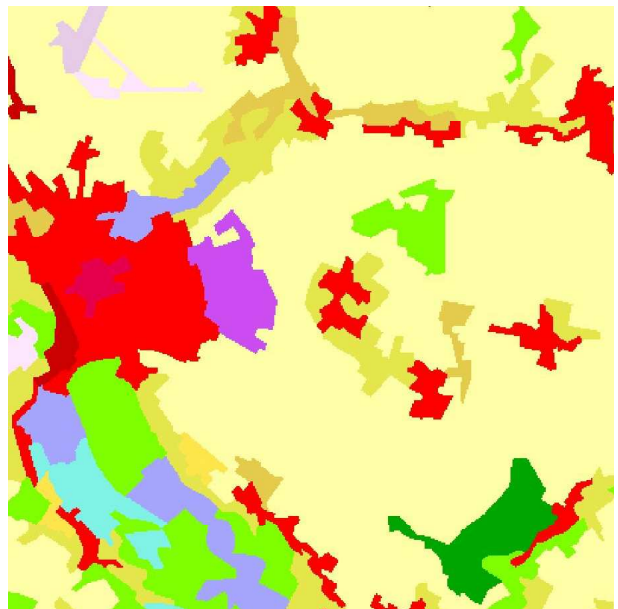


Figure 5. CORINE land cover

The first step to accomplish in order to determine if the land cover could really help us to enhance our automatic classification method was to check that it gave good results with manual classification.



Figure 6. Correlation between BD ORTHO® and CORINE land cover

Then we used the land cover to divide the aerial photo into layers in the SE-CLASSIFICATION meaning and used SE-CLASSIFICATION to simulate an automatic classification with a four colours palette in each layer.

The result was highly instructive.



Figure 7. Classified texture

As we can see it from the images above, the land cover layers limits can clearly be seen. Another problem is that the urban areas and the forest seem very uniform.

The layer limits problem can be obviously explained. First the resolution of the land cover is very low. The layer limits are thus raw. Besides, in the real life, the limits between a village and its

surrounding fields are not so clear.

The uniformity problem on urban areas can be explained by the fact that we used four materials by layer, and that is far from enough for an urban area, and that we used SE-CLASSIFICATION that interpolates materials according to the colour interpolation, and in an urban area, with four materials, this leads to a smoothing of all the area.

The uniformity problem on a forest can be explained by the fact that a forest has a fairly uniform colour so with our method it ends up with the same material and thus is uniform. But in real forests, trees have different heights, different varieties of trees, creating colour variations.

5.3. Next step

One way to work around the layer limits problem could be to create border zones (by giving a thickness to the layer limits for example) where a pixel would be tested against the palettes of the nearest layers and its material chosen with a smart random if needed.

Let us give an example of an urban zone surrounded by an agricultural zone.

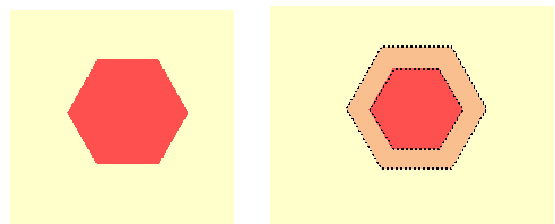


Figure 8. Urban zone with border area

We build a border zone by growing/shrinking the area limits.

In this area, we will test all the pixels against the two layer palettes. If only one material is close, we choose this one. If there two materials in the two palettes that are close, then we will choose the material randomly, according to the distance of the pixel to the layers: the closest we are of one layer, the higher is the chance to get its material. In the central area of the border area, the probability to pick one material or the other is equal.

The uniformity problem on a forest could be solved by adding bump mapping or random noise to create a modulation of the classified areas. But we must be very careful to keep a physical consistency and not be tempted to add a noise that has no physical sense because "it is prettier".

The uniformity problem on urban areas will naturally be enhanced by the automatic classification method since this method associates

one material to one pixel and does not interpolate materials according to the colours as SE-CLASSIFICATION does, but the main problem will be unchanged: an urban area contains a lot of different materials and, due to rendering software limitations, we will probably not be able to afford such a high number of materials.

6. ANOTHER LEAD: MIXING WITH TEXTURE SYNTHESIS

Using land cover and several palettes to enhance the automated classification could significantly improve the results, especially for urban areas. But it is a long way and it might be costly. And it will not solve the aerial photo resolution problem. When using Landsat images, a pixel is 16 m wide. Thus, when getting closer to the database, even if the classification is perfect, everything gets blurred (and we do not mean “pedestrian closer”, just “helicopter closer”).

A way to solve this problem and the uniformity problem on urban and forest area could be to add texture synthesis to the automated classified databases.

The principle is quite simple. First the Landsat aerial textures are automatically classified. Then, using land cover information, a detailed texture is synthesised, using generic classified textures representing forest, fields of various crops, even villages. The generic textures precision is around 50 cm. Those two textures are then combined using the multi-texturing mechanism, ensuring a smooth transition.

Experimentation was conducted on a Landsat image fragment, in the visible spectrum (the point here was to study the visual transition between the aerial photo and the synthesised texture) but it will be fairly similar in the infrared spectrum. The result of this experimentation is shown in the three images below. They represent the same database seen from a shortening distance: first from far away, then from a medium distance, then from a quite short distance.

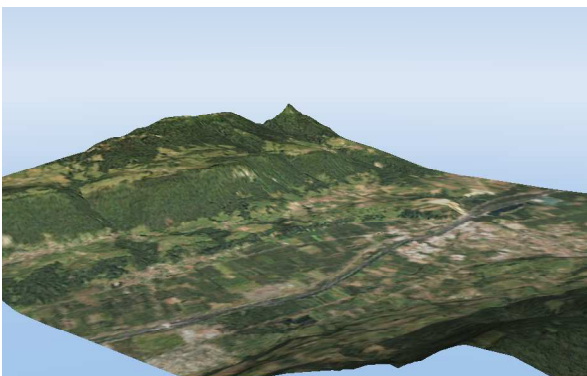


Figure 9. Database fragment seen from far away : the Landsat image is seen

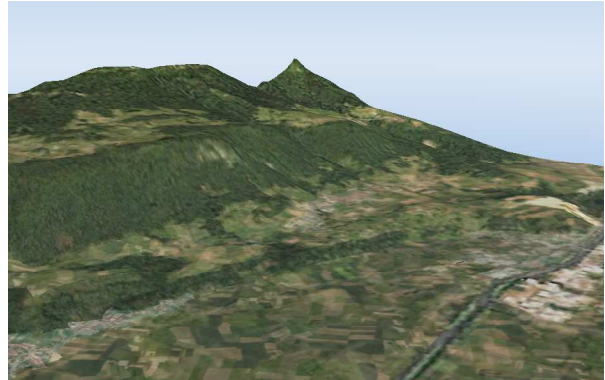


Figure 10. Database fragment seen from far away : the Landsat image is seen



Figure 11. Database fragment seen from medium distance : only the synthesised texture is seen

This method could also participate in solving the texture weight problem. When dealing with wide databases (several hundreds of kilometres by several hundreds of kilometres), especially for real time rendering, the amount of textures can be huge. And the limitation is clear: the global texture size must be inferior to 2Go. The following table sums up the memory cost for a million pixel texture, for a database that should be rendered in both the visible and infrared spectra, for the OKTAL-SE real time and non real time rendering products. For example, for a 128 km x 128 km database, we would need 360 millions of pixels, which makes a memory cost of $8,5 \times 360 = 3060$ Mo for eight materials...

Using Landsat 16 m precision images, the same area can be covered by a single texture of $8\,000 \times 8\,000$ pixels that weights $8,5 \times 64 = 544$ Mo for eight materials. When adding synthesised texture as a detailed texture, each different generic texture should have a size around one kilometre. For a 50 cm precision, this makes a one million of pixels texture weighting $8,5 \times 1 = 8,5$ Mo for eight materials.

It is thus clear that even if we use a dozen of generic textures (that should be enough for describing countryside areas), the global texture weight is much lower and the visual result is correct, even if we still have to work on the

transition between two generic textures (as shown in the image above, when the synthesised texture is seen from a short distance).

7. CONCLUSION

This experimentation showed us that the “rough” method results were not as bad as expected. They can even be considered as correct for aerial database that do not require a high precision level. Some ideas to make it more precise are simple (for example creating palettes with a lot of colours leading to a low number of materials in order to make sure that the colours are associated to the material we want) but even testing those ideas takes a lot of time.

So before thinking of setting up a more complex method using land cover or other information, we should first speed up the production process in order to be able to validate this method further and see where the simple improvements lead us.

Two steps represent major bottlenecks for the “rough” method: the water pre-processing and the palette creation.

Creating a palette requires to describe each colour of the palette with its RGB components and to find, in the multi-domains materials library, the logical material that will be associated to this palette.

The user has then to manually add all the logical materials used in the palette to the material file to classify, reference the multi-domains materials library used before the RGB2IPC tool can be started. This task can be long if we have to build a palette with many materials and it is tedious.

This bottleneck could be widened by a simple tool allowing the user to choose the multi-domains materials library to use, the material file to classify and to build the palette using a friendly GUI: creating a list of pairs, one being a RGB colour chosen from a colour. This tool could then suppress any previous classification or just classify the texture that has no classification yet, copy the logical material chosen, reference the multi-domains materials library and start the automatic classification.

We could also imagine that this tool could generate a benchmark 3D database, allowing the user to easily test the automatic classification with the OKTAL-SE rendering tools.

We could also imagine that this tool could be able automatically create a composite texture from all the textures to classify, reduce the number of colours of this composite to a given number and suggest a primary palette that the user can then modify. While we are at it, it could also create the

materials file to classify.

Another bottleneck is the pre-process of the water. This is an extremely long and tedious task, especially when dealing with wide database requiring several hundreds of aerial photos.

A major improvement could be to automate this process. But automating such a process is complex and will probably require simplifications.

For example, we could imagine using cartography information (land cover or vectorial information) to automatically make the rivers, lakes, etc. as water. This would of course lead to rivers that would be much more regular than the real rivers but, in the optic of database meant for aerial simulation, this should be acceptable and save a huge amount of production time.

This could also solve the “road problem”: if we are able to pre-process linear feature such as rivers and associate them right away with the correct material, we could do the same for roads.

8. REFERENCES

1. A. Le Goff, J. Latger, *Realistic multi spectral simulation including IR simulation. SPIE Proceedings.*, Vol. 3694, April 1999.