Toward a Generic Approach to Stylization

François-Xavier Talgorn
Université Paris 8
ftalgorn@ai.univ-paris8.fr

Farès Belhadj
Université Paris 8
amsi@ai.univ-paris8.fr

Vincent Boyer
Université Paris 8
boyer@ai.univ-paris8.fr

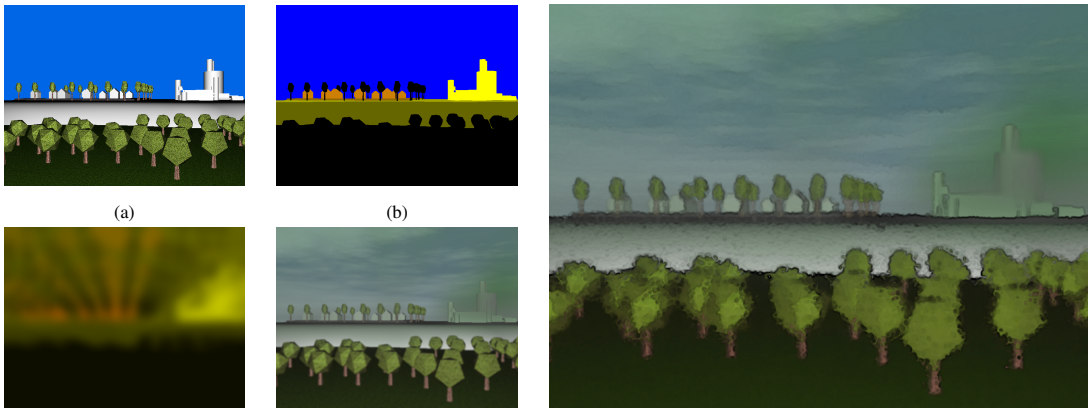


Fig. 1: Stylization: from basically shaded scene rendering (a) to semantic-expressive rendering (d, e) through semantization process (b, c).

***Abstract*—This paper presents a first attempt toward a generic process for stylization. Guided by the scene content and constrained by the artistic or functional expectations of the end-user, our model produces, in real time, an expressive rendering of the scene based on the user settings. Our work is based on semantic description of shapes that allows logical inference on the scene content, performed via the description logic formalism. This inference yields semantic data that can guide the graphical restitution of the scene. This approach is illustrated by a practical application example that successfully stylizes a 3D scene in real time and for a predetermined style. The resulting stylization, even if style is selected in advance, automatically and successfully highlights different parts of the scene according to semantic data extracted for each object. The stylization results are then discussed before we conclude and present our future works.**

I. Introduction

Up to now, the stylization problem has been addressed in a siloed approach. Almost all research works in artistic rendering field aims for the production of a specific style which is often efficient but non reusable outside of its specific scope. It should be noted that it is a common way to break down a problem into more specific subordinate aspects to propose partial solutions and work around a fundamental question. As an example, Kyprianidis has proposed a state of the art on stylization techniques and lists about thirty different families of expressive rendering processes for the sole image based artistic rendering type (IB-AR) [1]. The vast majority of those techniques tend to replicate a very specific visual style, e.g. oil painting, hatching or watercolor rendering. After more than thirty years of active research, the expressive rendering field yields a wide range of works that explains *how* to produce a graphic style while the fundamental questions of what a visual style *actually is* and *how it relates* to the content is never addressed.

Furthermore, as a general *functional* aspect of stylization, it is important for us to gain control over the information that will trigger the generation of a given style. If they were such a consistent generic model of visual style on which NPR works could lean on, this would move out the global research effort of siloed approaches and bring new opportunities in addressing today's artistic rendering challenges. Indeed, profound questions like 'What beauty is ?' or 'Is this quality work?' can not be answered without generic models that transcend all technical solutions and shift the problem from **how to** generate a graphical style to **what is** stylization, intrinsically.

Example Based Rendering (EBR) techniques, along with other global processing such as image filtering, can reproduce a wide variety of artistic depiction without any useful information on the style itself. We are still blind regarding its specificities and intrinsic properties.

Until we are able to define and manipulate structure and data that relate to style concepts - independently of any image content - we can not separate the content from its representation, making it difficult, if not impossible, to manipulate the countless visual styles in a *unified* manner.

The main contribution of this paper is to introduce an approach to stylization that goes beyond functional categories, e.g., physical simulation, algorithmic or statistic representation, in order to propose a consistent generic model for visual styles description. After discussing some expressive rendering techniques related to our work, we present the fundamental

ideas that underpin our work, detail our approach and how the resulting model provides new elements to move toward a formal, generic structuring of visual styles. Then we provide an example of 3D stylization, among many other practical applications that could be based on our model, before discussing the approach. Finally, we conclude and provide prospects of this work.

II. State of the art

In this section, we present related work on expressive rendering and the quest for graphical style definition. As mentioned in the introduction none of these works defines what style actually is. This kind of question is never addressed in research works reproducing visual style since the challenge is to compare computed image synthesis and artistic production. Only states of the art on expressive rendering approach this issue since they propose a taxonomy of related work. Therefore, considering the classification presented in the taxonomy, we extract concepts of graphical style definition.

Kyprianidis [1] recently surveyed image-based artistic rendering techniques (IB-AR) since the 80's. According to Kyprianidis taxonomy, IB-AR techniques applied to video or still images are divided into 4 main families: stroke-based rendering, region-based rendering, example-based rendering and image processing and filtering.

From early works on brush stroke simulation [2] to recent image parsing in order to extract semantic data on graphical contents [3], stroke-based rendering techniques have achieved remarkable results. These techniques succeed in reproducing many painting [3] [4], drawing [5] [6] or mosaicing aesthetics with great visual richness and natural feeling like in Orchard et al. [7] or Hurtut and al. works [8].

Region based techniques are commonly used as segmentation tools in expressive processes. Nonetheless, some works yield very interesting results based on region oriented processing like in the work of Wang et al. for toon shading [9], in graphical restitution of textile (felt) as in Donavan et al.'s work [10] or black and white image depiction with the work of Xu [11]. Image processing and filtering yields few interesting results as a stylization tool. This is due to the fact that these filters are mainly used for restoration and enhancement of photorealistic images. They are globally divided into two categories: spatial and gradient based. Most techniques are applied in the spatial domain while a sparse research effort is made in the gradient one. However, the latter may produce very interesting results like in Bhat and al.'s work [12] where their abstraction result gives better results compared to Ozran et al.'s [13] and Winnemöller and al.'s ones [14].

The last family of Kyprianidis taxonomy is the example based one. Since Hertzmann's work on image analogies [15], EBR pioneered a novel approach to style generation by not trying to reproduce a specific style, but rather learning an analogous transformation from a training image pair (a source image and an artistic representation of it). This transformation is then applied to any new image by the mean of the ad-hoc mathematical operation. Some interesting recent works not only transfer colors but also brush strokes from a dictionary of templates to literally *paint* by analogy [16].

None of these works propose a style definition, yet they share at least a core concept: to generate a style, additional information is necessary. This information is included in the scene, e.g., an image, a video, a 3D scene and most of research works extract it either automatically, semi-automatically or manually. Thus, in order to provide this higher-order information for stylization purpose, we propose a semantic description of shapes. This semantic layer allows for a logical inference of the scene which makes possible a semantic tagging of its content. The latter can be used to guide a graphical restitution of the scene.

III. Our model: a semantic shape generation

Our model aims at addressing the question of style itself, not the production or reproduction of a particular style. To achieve this, we seek to characterize all elements present in a scene. A graphical restitution of those characteristics would allow to produce an image. Our model should make it possible to choose different graphical restitutions for a specific characteristic, potentially yielding different graphical styles. We choose the form as the core element on which applying a graphical style.

In order to be able to manipulate the concept of a form - before its representation by a given style - we *semantically* define it using a reduced set of geometrical concepts and a corresponding set of volumetric primitives associated with them. In our approach an object is described by its *ontological properties* - in the sense of the irreducible *being* criteria. For example, our mental representation of a car is associated with the concepts of wheels and passenger's compartment, which reflects the reality that cars always have wheels and a passenger's compartment. The number of wheels may vary, as for the size of the compartment but every single car shares those two ontological properties. Hence, representing a three dimensional car with the help of some torus, disks or cylinders (the wheels) and a cubical volume (the passenger compartment) along with its average spatial dimensions would be enough to describe and identify a class of objects we name `cars`. Building such semantic knowledge about forms provides a way to describe and identify them regardless of their graphical representation.

This approach is inspired by the way our visual cortex processes external visual inputs. From core electrical visual inputs, dedicated groups of neurons identify basic shapes that are gradually aggregated into a form we recognize and eventually name with the help of our memory. These basic shapes, e.g., dots, line, angles, act as descriptors (among others) with which our brain tries to match the closest known object. Elementary forms and other data such as color, position or light intensity constitute together a set of descriptors. The aggregation of descriptors leads to the identification of new descriptors (at higher semantic level) such as the object topology. Similarly, in our work, taken together, these descriptors allow for object recognition by matching them to a database

of semantic description of objects and provide the scene with a first level of semantization. For obvious reason related to the size of the database, the latter knowledge base of objects can be created/updated by the final user depending on his needs regarding the type of content to be analyzed. From there, an inference can be performed on this semantic base and enrich our scene knowledge. By defining basic shapes and associated semantics, the scene *characteristics* are define and a graphical restitution can be applied. Therefore, this generic model can be applied by defining the *shapes*, the *semantics* and the *graphical restitutions* associated with them.

The binding between actual shapes and their formal semantic representation can be achieved by an axiomatization process based on description logic (DL), as introduced by Krötzsch et al. [17]. The main advantage of this approach is that description logic can be inferred in a decidable way when expressed with restricted predicate logic, assuring the sound and complete logical inference. With this formalism, it becomes possible to compute formal logical relations on high level conceptual elements such as shape, color or orientation. This kind of formalism has been used by Ben Hmida et al. [18] in the form of SWRL language to identify specific objects from a 3D scan.

Our approach proposes a taxonomy of volumetric primitives and a set of logical relations. Together, they constitute a blueprint for object definition and inference. Depending on user needs, objects classes and properties are defined, as well as higher semantic classes that rely on the previous ones, i.e., primitives shapes define object classes which can be used to define higher semantic entities. As an example, depending on their density in the scene, cars and houses classes (defined by volumetric primitives) may define a city (a higher-order entity defined by classes). Hence, by defining dedicated classes and properties based on the taxonomy and the associated formalism we provide, the user can build (or update) a semantic representation of objects associated with a domain, e.g., architecture, weaponry, scenery, and use it to infer scene characterization, as shown on the left side of figure2. Then, following the process of inference, semantic data are available and can be used to enrich the initial scene and guide its stylization for expressive rendering purpose, as shown in the right side of figure 2 and detailed in figure 3.

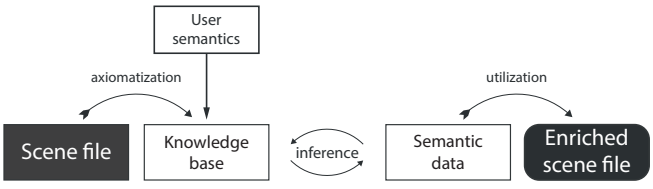


Fig. 2: Main steps of scene enrichment by axiomatization process.

IV. Application example: semantic-based rendering

As an illustration of our purpose, we propose an example with a 3D scene that contains natural and industrial elements (trees, houses and a factory). Our model is used to semantically define this set of objects. This semantic definition then allows for the automatic inference of their nature (a tree, a house). Based on the result of this inference, two types of semantic properties are associated: an object can be *Pollutant* or *Purifying* and exudes an atmosphere that ranges from *Reassuring* to *Creepy*. Eventually, these semantic properties are used to guide the graphical restitution of the scene. As shown in figure 3, an abstracted version of the original scene is built. This abstraction layer could be built upon Yumer and al. work [19] and Mehra and al. [20] results. In the present case, in order to focus on the semantic part of the proposed approach, abstracted parts of objects are tagged semi-automatically with their volumetric primitive types, e.g., *plane*, *solidCubical* or *solidSpherical*. This scene is then automatically axiomatized as described in section IV-A. The resulting inference allows to potentially identify an object as being the ground, a tree, a house or a factory. Based on this semantic data, two types of parameters are set for each object that has been identified: its pollution factor and the atmosphere it provides. The level of pollution ranges from *Purifying* (which suits well for a tree) to *Pollutant* (which is the case for a factory). The atmosphere ranges from *Reassuring* to *Creepy*. Both pollution and atmosphere parameters act as inputs to guide the graphical restitution in real time by adding smog and adapting the lighting of the scene, respectively.

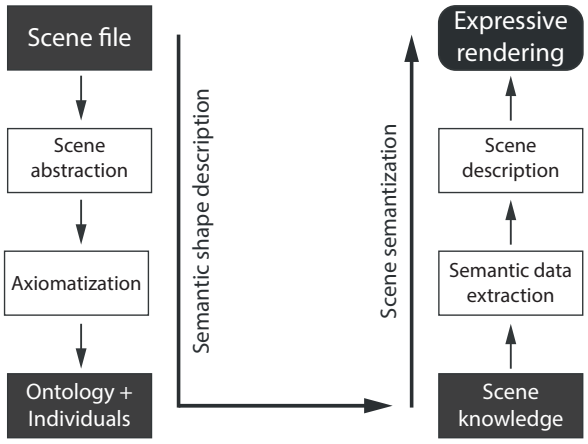


Fig. 3: Pipeline overview. From a 3D scene file, objects topologies are translated into an ontological representation. An inference on the resulting knowledge base provides semantic information such as object identification and scene tagging. The latter being used in our case for stylization purpose.

A. Semantic shape description

In this example, the semantic knowledge associated with any object in the scene is defined by three ontological properties:

- the number and type of its constitutive parts;
- its overall actual dimensions;
- its global orientation in space.

The axiomatization of the abstracted 3D scene is realized in OWL-DL language (Ontology Web Language in its DL form) which is a fully decidable version of OWL. It is a *SHOIN(D)* description logic that provides datatypes (literals that can take numerical values) useful in our case to set and infer on actual size of objects. The constitutive parts of 3D objects, i.e., the volumetric primitives that constitute the abstracted representation of an object, are taken from a subset of *geons* as defined by Biedermann [21]. This choice is guided by the fact that it is of no use to define new and dedicated 3D primitives in the context of our example scene. In OWL-DL terminology, these volumetric primitives are defined as top classes (ex. `solidCubical`), the overall dimensions of objects are defined as datatypes (ex. `hasMaxBoundingBoxValueOf`) and their global orientations in space are defined by two data properties (`isHorizontal` and `isVertical`). The purpose of this axiomatization is to build an ontology that *semantically* describes the scene (left side of figure 3) in order to be able to perform logical inferences that, hopefully, produce the identification of objects present in it (right side of figure 3). The resulting identification of objects is finally used to set their *Purifying*/*Polluting* and *Reassuring*/*Creepy* parameters that will guide the graphical restitution of the scene.
The ontology produced for this example is twofold: a taxonomy of volumetric primitives and classes that use this taxonomy to logically define objects based on our three ontological properties. For the purpose of this example we define four classes of objects: *ground*, *tree*, *house* and *factory*. These objects are defined by the use of the taxonomy and a `isComposedOf` relation that reflect the relation of an object to its constitutive parts. Furthermore, several properties for the size and orientation in space of object are respectively defined as a range of values in centimeters, and `isVertical` , `isHorizontal` and `isEven` as boolean values. As an example, the abstracted representation of a common tree can be defined by one cylinder for the trunk, and a rather spherical or conical volume for the foliage. Its size may range from 2 to 25 meters if very tall trees like old sequoias are ignored. We can formally describe this abstracted, semantic representation of a common tree by the following OWL description:

```
isComposedOf only (SolidCylinder or
SolidSpherical or SolidConical)
and isComposedOf exactly 1 SolidCylinder and
hasBoundingBoxMaxValueOf
(some double[>= 200.0] and double[<=2500])
and isVertical=true.
```

Once the ontology with its taxonomy of volumetric primitives and logical description of objects classes is generated, the 3D scene has to be translated in this OWL-DL formalism in order to update the ontology with *actual* objects present in the scene. In OWL-DL terminology, these 3D objects will become *individuals* (instantiations of one of our four object classes with actual values). In order to instantiate these individuals, we first extract useful data from the 3D scene, translate it with the OWL-DL formalism in our ontological representation of objects, and finally update the ontology file with the resulting OWL-DL logical transcription. Data extracted is:

- the ID of the object;
- the name of its parts;
- both local and global transformation matrices;
- the 3D points list of each of its parts.

From primitives' names, 3D points lists and transformation matrices, additional information is extracted or calculated:

- the number and types of volumetric primitives that constitutes the object;
- for each primitive, its three spatial dimensions (height, width, depth);
- the three spatial dimensions of the global bounding box;
- the global orientation of the bounding box (`Vertical, Horizontal or Even`) .

With this data, a knowledge base that reflects the content of the 3D scene can be created. The instantiation of each *individual* is two-steps: assert the topology of the object and set the values of datatypes and data properties for its size and orientation. The topology of objects, defined in this work by the type and numbers (cardinality) of its primitive components, translates into the following logical description: the intersection between all values of the union of components and the `isComposedOf` relations of cardinality n, for each component.

Following this axiomatization, the ontology is updated with as many *individuals* as matching objects in the 3D scene. Once the ontology is populated, a semantic reasoner that classifies the individuals among the top classes is used (we choose the Pellet reasoner for its performance). The resulting inference yields a tagging of the individuals. This semantic data is then used to enrich the original 3D scene. In this application example, objects can be tagged as *tree*, *ground*, *house* and *factory*. Based on the result of this tagging and the unique ID of objects, the initial scene can be enriched with high level semantics: for each object identified in the scene, a value for the *Polluting* and *Creepy* factors is set. As shown in figure 4, these values are encoded on two channels of a material property associated with each 3D object. These two *Pollution* and *Creepy* factors range from 0 to 1 and reflect for each object the diffusion level of these two properties. Thus, this semantic knowledge can generate a semantic map used to stylize the 3D scene in real time. This stylization process is described in detail in the next section.

B. Semantic-based rendering

Our rendering process needs the geometry of 3D objects, normal vectors for lighting, objects materials, texture coordinates, the textures and the raw semantic weightmap given as a texture, as shown in figure 5(a) and (b). In this case, only the red and green components are used to store two types of semantic data that reflect the *Polluting* and *Creepy* factors diffused by each object of the scene, as shown in figure 4.

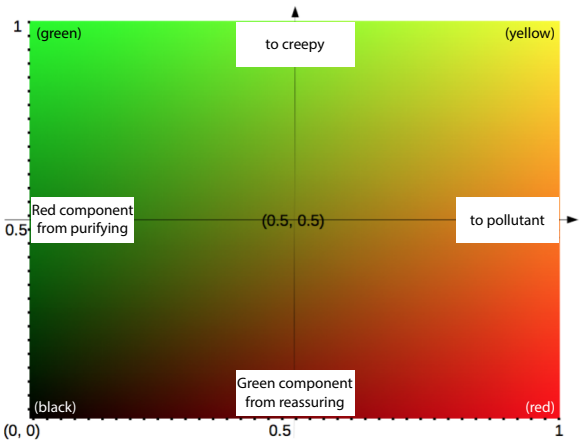
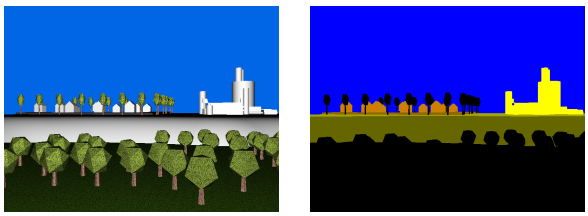


Fig. 4: Semantic data representation along two axis and four different categories. The two channels used to describe pollution (from *purifying* to *pollutant*) and atmosphere factors (from *reassuring* to *creepy*) are represented with red and green color, respectively.

Based on the tagging of objects, a specific color property in objects' materials is used to encode the level of pollution and the type of atmosphere it exudes. As shown in figure 4, the red component is used to set the level of pollution while the green component sets the exuded atmosphere. Along those two axis, four components can be set. Both opposite directions of each axis (the *Polluting* and *Creepy* axis) are coded in two intervals $[0, 5[$ and $[0.5, 1]$ in order to fit the OpenGL/GLSL texture components' range of $[0, 1]$. Here the blue component is used to denote the absence of any semantic information. The resulting semantic map is shown in figure 5(b).



(a) (b)

Fig. 5: Raw scene (a) and semantic map generated from the results of the axiomatization mapped on red and green components (b). Blue component denotes the absence of semantic data.

In this raw semantic map, visual transitions between the different zones of the resulting image are very abrupt and could produce visual artifacts. Thus, the semantic colorization has to be smoothed and diffused so that the corresponding rendering produces soft visual transitions between the two rendering axis (pollution and atmosphere). Furthermore, the choice of the diffusion method should be guided by the constraints of expressive real time rendering. A 2D gaussian blur being implementable as a composition of two 1D gaussian blurs (a composition of a horizontal and a vertical blur), this solution can be particularly efficient in GPU to allow real time processing even with large convolution matrices (more than 512×512 kernels).

However, the question of elements that are not associated with semantic data has still to be addressed. This data is tagged with blue color in the semantic map as shown in 5(b). One solution shown in 6(a) is to set the semantic data to *neutral* when the corresponding data is lacking (a value of 0.5 for both *Polluting* and *Creepy* parameters). The result shown in 6(a) after a gaussian blur with a 65x65 kernel would be satisfying but does not produce a good result when a large part of the scene is not semantically tagged. We can notice in this case that semantic data has a very little impact on the sky (or everything else if the scene was different). In order to overcome this problem, we propose to generate semantic data in a consistent manner and use it to fill the empty zones of the semantic map. This can be achieved by using the intermediate result shown in 5(b). In this case, the semantic map is stored in a texture via a framebuffer where the blue component is considered as an alpha factor. Furthermore, we calculate the center of the scene in screen coordinates. We then utilize this data to draw several occurrences of the texture at different logarithmically decreasing scales centered on the center of the scene, while cumulating the alpha transparency and applying a blend constant factor of 2/number of scales. This yields the result shown in 6(b) which is then blurred as shown in 6(c).

The latter result could be sufficient but does not provide an easy way to fine-tune the contrast between opposite semantic axis by applying different weights. This is due to the fact that opposite directions are stored on the same color component. Thus, when a weighting is applied before the blur filtering, one can only expect a slight enhancement of contrast between the two semantic data. On the other hand, if weighting is applied after the blur filtering, visual discontinuity would arise. To overcome this issue, we propose to demux the values by splitting (two for each semantic axis) and store them on the four components of the RGBA texture. By doing so the red component would keep a neutral or *Polluting* value while copying a *Purifying* value into the blue component, which leads to:

$$\texttt{if } (R < 0.5) \ \{ \ B = 1 - 2 * R; \ \ R = 0; \ \}$$
$$\texttt{else } \{ \ R = 2 * (R - 0.5); \ \ B = 0; \ \}$$

The same operation is performed on the green component, using the alpha one to dispatch its opposite value. This *demuxing* operation allows us to perform any operation on the RGBA texture without any dependency between opposite semantic values (such as *Purifying* vs. *Polluting*). Once the operations are performed we *remux* the four values by the inverse calculation process. As shown in 6(d), semantic data are *demuxed* then weighted independently before applying a blur in one pass on the whole texture. In practice and in this example the polluting component is kept identical (linear) while the *Purifying* one is bumped by applying a power factor in $]0, 1[$. The four resulting components are given in 6(e) with $black = 1$ and $white = 0$ for the negative version. The image

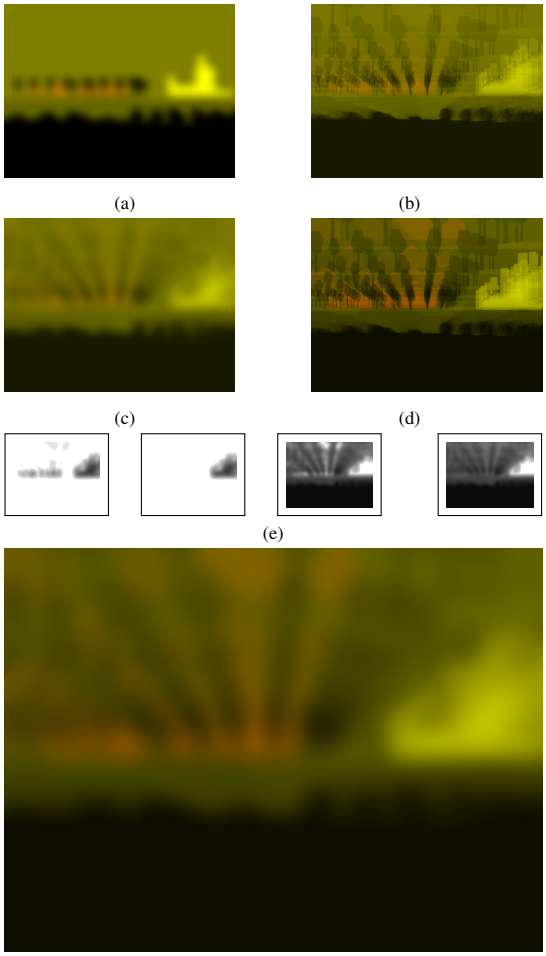


Fig. 6: Real time processing of semantics for immediate use by the rendering engine.

at the bottom of figure 6 illustrates the four components after *remuxing* and shows a higher contrast. This latter semantic representation of the initial scene can be used directly for real time processing. In the next section, we propose expressive renderings based on semantic data modified this way.

V. Result and discussion

With only a few semantic data (four in our case), some 3D and post processing effects can be applied to produce interesting images. In this application example we seek to highlight the effect of pollution on the environment while producing a visually interesting rendering starting from a raw 3D scene. By using both the *Polluting* and the *Creepy* factors associated with objects, post-processing effects such as smog and specific lighting are generated. Furthermore, in order to visually enrich the graphical restitution, 3D processes are applied to produce a painterly rendering of the scene. In the first example, the texture of the sky (clear or cloudy) is chosen in accordance with the global pollution level while the pollution levels generated by each of the polluting objects is visually represented by adding more or less smog in affected areas. A green creepy lighting is also generated around objects according to their atmosphere factors, i.e., value of green component greater than 0.5, as shown in figure 7.

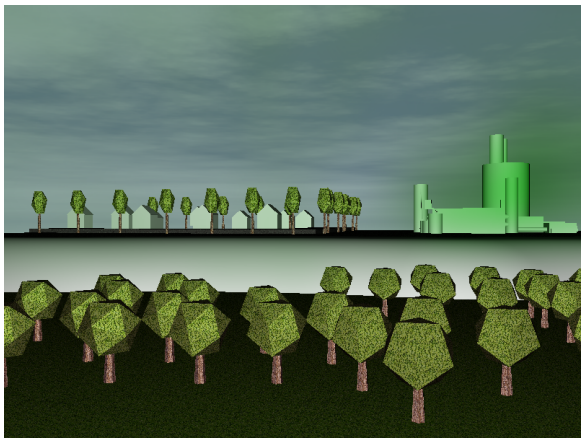
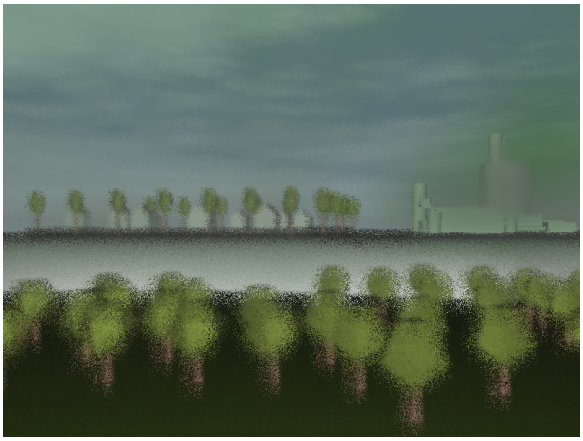


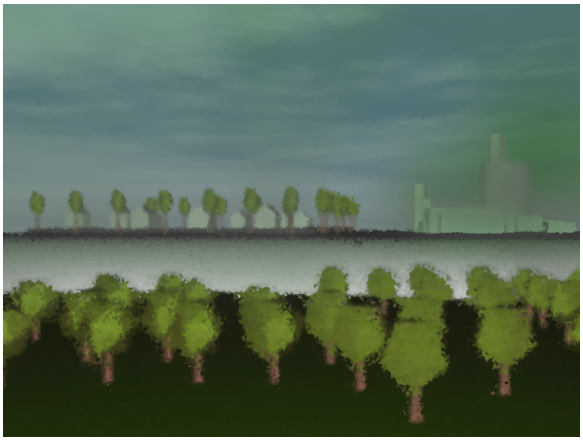
Fig. 7: The original scene is enriched with a sky texture that depends on the pollution level of the scene, while a green creepy lighting reflects the detrimental effect of polluting objects (factory).

While the visual effect produced by this simple post-processing is interesting to render the different objects in the scene according to their natures (e.g., a tree, a factory) the semantic weightmap can be used to guide the rendering of the scene on an expressive, artistic level. To illustrate the potential of expressive rendering based on semantic data we choose to apply different filters based on the semantic map, producing different visual aesthetics according to our semantic coding (*polluting*/*creepy*). Here, we want to give a negative feeling about polluting objects. Thus, polluting objects are heavily blurred in order to reduce their contours detection by the associated edge detection process, which eventually dilute them in the smog effect. By contrast, natural elements should yield a positive feeling. In this example we choose to represent these positive entities with a painterly rendering. This is achieved by scattering the pixels on the *Purifying* areas before applying a median filter to visually unify the result. In order to increase the impressionist style regarding plants, the higher the *Purifying* value is assigned, the more scattered they are. This allows for the following median filter to produce a stronger brush strokes effect. Also, we apply a light blur on the objects tagged as *Purifying* to avoid too many resulting brushstrokes. In a nutshell, these choices produce an impressionist rendering on trees and ground while a soft contrast between polluting and regenerating objects is visible, as shown in figure 8.

Finally, to further increase the abstraction contrast between natural elements and the rest of the scene we choose to add a sobel filter which strengthens these natural elements. The final GPU rendering is shown in figure 9 while the whole process is illustrated in figure 1. The latter is produced on a *Nvidia Quadro FX3800* and a 2.46 Ghz *BiXeon* CPU processor. The scene used in this application example is composed of 22K



(a)



(b)

Fig. 8: A scattering operation (a) and median filter (b) applied to objects tagged as `natural` yields a strong impressionist feeling.

mono-textured triangles and rendered at 800×600 resolution. We obtain:

- 475 fps for a basic Phong rendering;
- 155 fps when only computing the semantic weightmap (see figure 6);
- 80 fps for an expressive rendering without the impressionist style (scattering/median);
- 15 fps for the final rendering, low frame rate mainly due to the three pass median filter that is applied to unify the scattering of pixels applied to trees.

In order to illustrate an alternative use of the semantic data, we propose to highlight the detrimental effect of the polluting objects on the environment. One way to do this is to simply invert the sense of the *purifying/polluting* axis (figure 4). This produces a totally different visual result as shown in figure 10 where the factory is highlighted. Furthermore, we have tested our model in another case where the expressive graphical restitution is purely functional (as opposed to a stylized rendering). In this case, we expected the system to colorize an interior scene depending on the type of objects

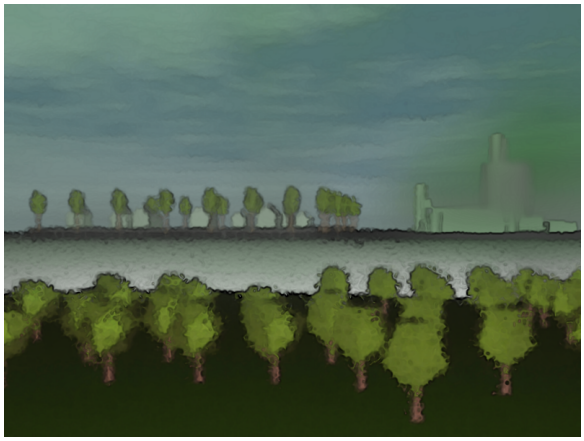


Fig. 9: The final semantic based rendering.

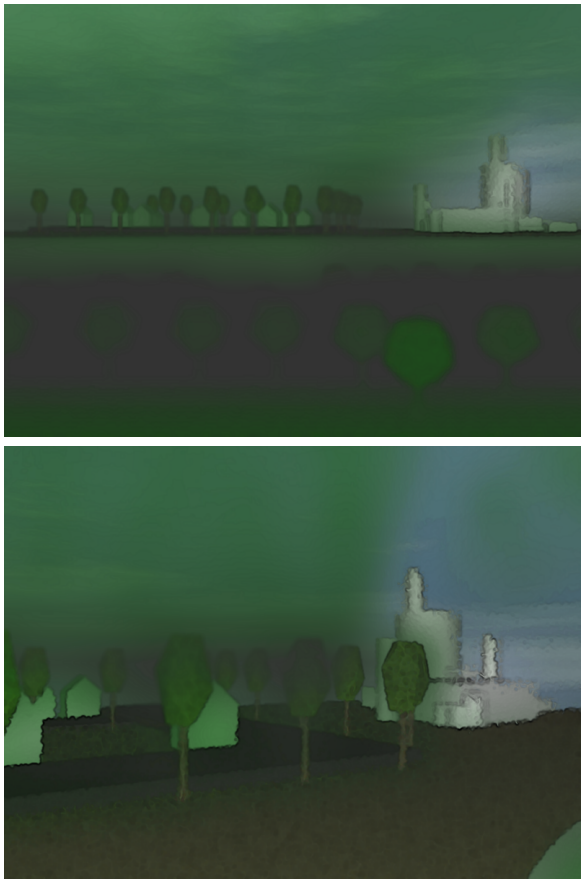


Fig. 10: Two different points of view of the same scene with the use of an inverted semantic map.

present in the room. As shown in figure 11, objects are successfully identified based on their semantic descriptions.

VI. Conclusion and perspectives

We have presented a generic approach to stylization based on a semantic description of shapes. Our model allows for the logical inference on the content of a given scene. The result

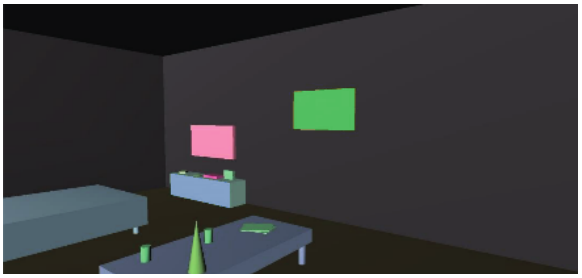


Fig. 11: An abstracted 3D scene where everyday items (glasses, vase and magazines on the coffe table, painting on the wall), furnitures (coffe table, console, bed), equipment (TV, Dvd player) and architectural elements (walls and floor) have been successfully tagged and colorized in green, blue, magenta and grey, respectively.

of this inference yields semantic data that enrich the initial scene and can be used to perform expressive rendering. We also have presented an application example based on the model to stylize a 3D scene in real time. This example successfully illustrates the capability of our approach to generate useful semantic data and guide the graphical restitution of a 3D scene both for artistic and functional use. In our case, a semantic weightmap is used to apply in real time 3D and post-processing effects that highlight the detrimental effects of polluting objects, automatically tagged by the system.

Our future works include a description language that would allow for dynamically set parameters defining the styles (in our example: scattering of pixel, blur, sobel and median filters, and automatic texturing of the sky). We could imagine that these parameters and their priority order could be defined via a description language. The description itself could be automatically generated or constrained by the end-user depending on his needs or artistic expectations. The semantic description of these parameters could be a general artistic direction (ex. oil painting, hatching) or precise guidelines for the final rendering like highlighting one aspect of the scene or managing the visual persistence of some elements in an animation.

REFERENCES

[1] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, "State of the art: A taxonomy of artistic stylization techniques for images and video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 5, pp. 866–885, May 2013. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2012.160

[2] S. Strassmann, "Hairy brushes," *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 225–232, Aug. 1986. [Online]. Available: http://doi.acm.org/10.1145/15886.15911

[3] K. Zeng, M. Zhao, C. Xiong, and S.-C. Zhu, "From image parsing to painterly rendering," *ACM Trans. Graph.*, vol. 29, no. 1, pp. 2:1–2:11, Dec. 2009. [Online]. Available: http://doi.acm.org/10.1145/1640443.1640445

[4] J. Lu, P. V. Sander, and A. Finkelstein, "Interactive painterly stylization of images, videos and 3D animations," in *Proceedings of I3D 2010*, Feb. 2010.

[5] H. Li and D. Mould, "Structure-preserving stippling by priority-based error diffusion," in *Proceedings of Graphics Interface 2011*, ser. GI '11. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society, 2011, pp. 127–134. [Online]. Available: http://dl.acm.org/citation.cfm?id=1992917.1992938

[6] E. Kalogerakis, D. Nowrouzezahrai, S. Breslav, and A. Hertzmann, "Learning hatching for pen-and-ink illustration of surfaces," *ACM Trans. Graph.*, vol. 31, no. 1, pp. 1:1–1:17, Feb. 2012. [Online]. Available: http://doi.acm.org/10.1145/2077341.2077342

[7] J. Orchard and C. S. Kaplan, "Cut-out image mosaics," in *Proceedings of the 6th International Symposium on Non-photorealistic Animation and Rendering*, ser. NPAR '08. New York, NY, USA: ACM, 2008, pp. 79–87. [Online]. Available: http://doi.acm.org/10.1145/1377980.1377997

[8] T. Hurtut, P.-E. Landes, J. Thollot, Y. Gousseau, R. Drouilhet, and J.-F. Coeurjolly, "Appearance-guided Synthesis of Element Arrangements by Example," in *NPAR 2009 - 7th International Symposium on Non-Photorealistic Animation and Rendering*, ser. Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering (NPAR). New Orleans, LA, United States: ACM, Aug. 2009, pp. 51–60, paper session: Emulating the masters. [Online]. Available: https://hal.inria.fr/inria-00391649

[9] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen, "Video tooning," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 574–583. [Online]. Available: http://doi.acm.org/10.1145/1186562.1015763

[10] P. O'Donovan and D. Mould, "Felt-based rendering," in *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering*, ser. NPAR '06. New York, NY, USA: ACM, 2006, pp. 55–62. [Online]. Available: http://doi.acm.org/10.1145/1124728.1124738

[11] J. Xu and C. S. Kaplan, "Artistic thresholding," in *Proceedings of the 6th International Symposium on Non-photorealistic Animation and Rendering*, ser. NPAR '08. New York, NY, USA: ACM, 2008, pp. 39–47. [Online]. Available: http://doi.acm.org/10.1145/1377980.1377990

[12] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, "Gradientshop: A gradient-domain optimization framework for image and video filtering," *ACM Trans. Graph.*, vol. 29, no. 2, pp. 10:1–10:14, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1731047.1731048

[13] A. Orzan, A. Bousseau, P. Barla, and J. Thollot, "Structure-preserving manipulation of photographs," in *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering*, ser. NPAR '07. New York, NY, USA: ACM, 2007, pp. 103–110. [Online]. Available: http://doi.acm.org/10.1145/1274871.1274888

[14] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," in *ACM SIGGRAPH 2006 Papers*, ser. SIGGRAPH '06. New York, NY, USA: ACM, 2006, pp. 1221–1226. [Online]. Available: http://doi.acm.org/10.1145/1179352.1142018

[15] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 327–340. [Online]. Available: http://doi.acm.org/10.1145/383259.383295

[16] M. Zhao and S.-C. Zhu, "Portrait painting using active templates," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, ser. NPAR '11. New York, NY, USA: ACM, 2011, pp. 117–124. [Online]. Available: http://doi.acm.org/10.1145/2024676.2024696

[17] M. Krötzsch, F. Simancik, and I. Horrocks, "A description logic primer," *CoRR*, vol. abs/1201.4089, 2012. [Online]. Available: http://arxiv.org/abs/1201.4089

[18] H. Ben Hmida, C. Cruz, F. Boochs, and C. Nicolle, "From 3D Point Clouds To Semantic Objects An Ontology-Based Detection Approach," *ArXiv e-prints*, Jan. 2013.

[19] M. E. Yumer and L. B. Kara, "Co-abstraction of shape collections," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 166:1–166:11, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366185

[20] R. Mehra, Q. Zhou, J. Long, A. Sheffer, A. Gooch, and N. J. Mitra, "Abstraction of man-made shapes," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 137:1–137:10, Dec. 2009. [Online]. Available: http://doi.acm.org/10.1145/1618452.1618483

[21] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological Review*, vol. 94, pp. 115–147, 1987.