Stylization of shadows in ski-panorama maps : A case study on Atelier Novat

Anmol S. Hanagodimath

September, 2020

TU Delft

Inria

Stylization of shadows in ski-panorama maps :

A case study on Atelier Novat

Anmol S. Hanagodimath

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

Student number:	4736133		
Thesis committee:	Prof. Dr. Elmar E. Eisemann,	TU Delft,	supervisor
	Dr. Ricardo Marroquim,	TU Delft,	committee member
	Dr. ir. J.C.A van der Lubbe,	TU Delft,	committee member
External supervisors:	Dr. Joelle Thollot,	INRIA,	supervisor
	Dr. Romain Vergne,	INRIA,	supervisor

Thesis was completed in collaboration with:

Computer Graphics and Visualization group,	TU Delft,	Netherlands
MAVERICK team,	INRIA,	France

Abstract

Panorama maps can be defined as aerial-view paintings of geologically complex landscapes that are represented in a convenient manner to non-expert viewers. This thesis focuses on a specific variant of panorama maps known as "ski-panorama maps" or "piste-maps" drawn by the french landscape and panorama artist, Pierre Novat. These maps usually contain a multitude of visual cues that are specifically emphasized upon by the artist in order to improve viewer perception. Among these visual cues, cast-shadows play an important role in recognizing the shape, depth and height of the terrain.

Therefore, the main goal of this thesis is to understand the use of shading based on the underlying rule-sets created by the artist, Pierre Novat. Through these rule-sets, we propose two contributions : a brief study of the artistic style of Atelier Novat along with a rendering framework for shadow stylization. This framework is mainly based on image-space filters, that takes a set of DEM (Digital Elevated Model) files as input and produces an image that resembles a ski-panorama map.

The pipeline deals with the following components : a novel method for stylized shading - a set of methods for modifying structural properties of cast-shadows (according to artist-driven rulesets) and a hue transfer-function. As a final step, validation of the rendered images is empirically done by cross-referencing it with the specific panorama art-work and by taking feedback from the artist, Arthur Novat.

Acknowledgments

These past three years have been quite challenging, with soaring highs and equally crushing lows. I would like to extend my gratitude to all those who have supported me thus far. I would like to specially thank my supervisor, Prof. Dr. Elmar Eisemann, for his patience, critical remarks and valuable insight as well as entertaining all my questions, which may have been a tad silly at times! I would like to thank Dr. Joelle Thollot and Dr. Romain Vergne for their critical guidance, warm hospitality and helping me fit in with the culture and atmosphere at INRIA-Grenoble.

I would like to thank my friends and colleagues; Sunrise Wang, Vincent Tavernier, Maxime Isnel and Hugo Rens, from the hilarious tea-time discussions to helping me with graphics and OpenGL. Morgane Gerardin, Ronak Molazem and Alban Fichet for their advice on material appearance and physically based rendering. Dr. Fabrice Neyret, Dr. Nicholas Holzschuch, Dr. Cyril Soler and Dr. Georges-Pierre Bonneau for their help and advice on my thesis topic. Arthur Novat for clarifying my doubts and explaining his perspective on the panoramas.

Lastly but certainly not the least, I would like to thank my parents for their unconditional love and support, without which, I probably would not have reached this far.

Contents

List of Figures

1 Introduction

Representation and illustration of real world geographic data is an ongoing research topic within the fields of computer graphics and Geographic Information Systems (GIS) that sees it's application in a wide-variety of usecases ranging from open-world games, automotive simulation software, geographic-visualization to real-time cartography and mapping solutions. Therefore, the illustration of raw-geological information (in the form of multiple file formats : DEM, SDTS, ESRI etc. [1]) has become an important topic in cross-disciplinary areas. In such situations, artistic techniques can be explored to emphasize and model key terrain features which might otherwise be unnoticeable to the untrained eye. This thesis aims to explore a specific usecase of artistic-terrain depiction colloquially referred to as 'panorama maps'.

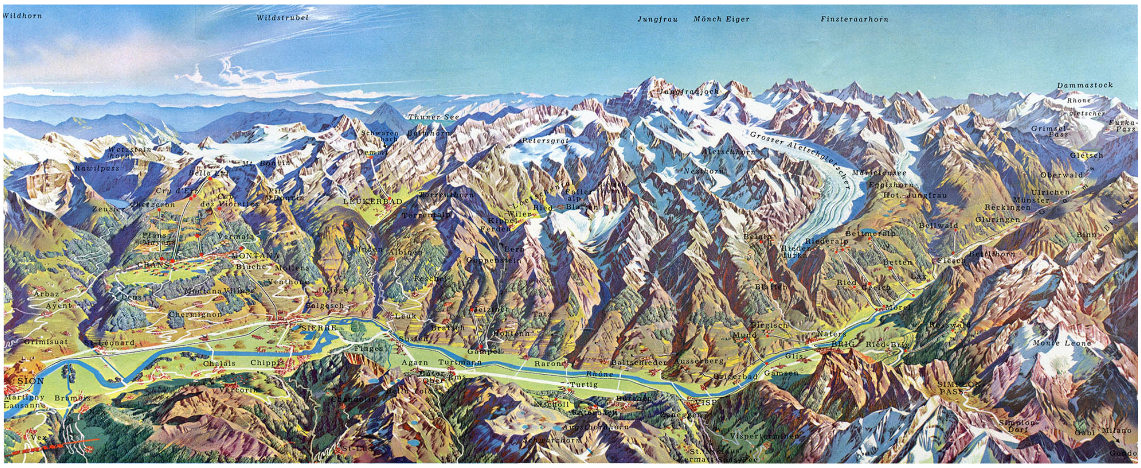


Figure 1.1: A summer panorama of Wallis by H.C. Berann, 1961

1.1 Panorama maps

Panorama maps can be defined as aerial-view paintings of complex landscapes that convey topographical information in a convenient manner to non-expert viewers. These panoramas form a bridge between art and conventional cartography in the sense that, artists responsible for painting panoramas are adept in cartography and have a keen sense of spatial awareness. Over the past century, a number of iconic panorama styles have surfaced that not only focus on seasonal changes but also lay emphasis on *how* terrain information is conveyed to the observer (Figure 1.1 and Figure 1.2 are good examples of contrasting seasonal panoramas; Chapter 2 delves further into artistic nuances between panoramas).

[1] http://www.reliefshading.com/

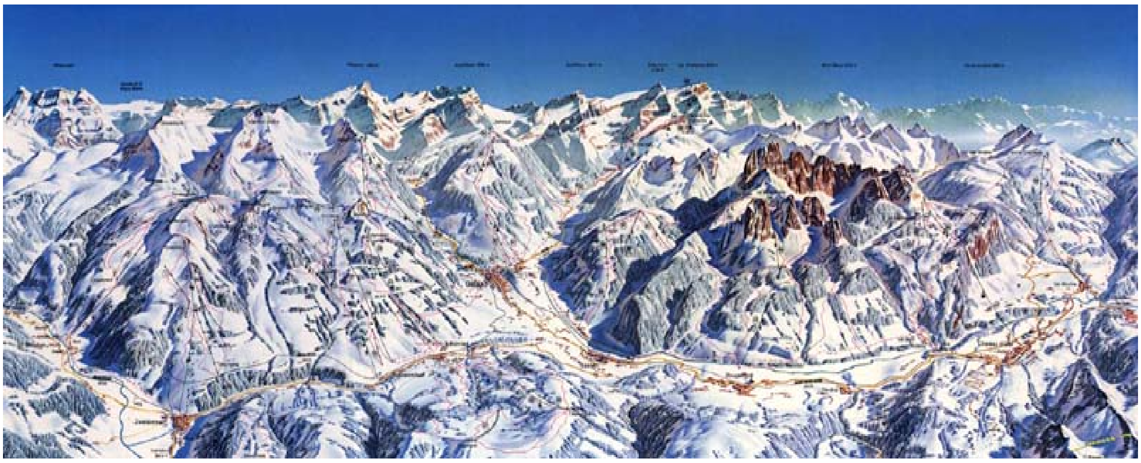


Figure 1.2: A winter panorama of Gstaad by H.C. Berann, 1986

The problems however, arise in the form of time constraints and the lack of automatic toolsets to facilitate the ease of drawing these panoramas. Since the area of expertise is high and the topic is fairly niche, there is a significant lack of data available for research purposes. Considering the fact that each panorama takes months to produce (from inception to evaluation from external committees to reiteration), it isn't surprising to note that most artists in this profession do not produce a large number of panoramas in their lifetime. For this reason, documenting key artistic considerations (emphasis on shading, shadows, terrain detail or surface texture to name a few) and automating this process becomes an important research topic. Moreover, this topic provides an interesting intersection between GIS understanding of cartography and artistic rendering of terrain.

In this thesis, the focus would primarily be on a singular aspect of panorama maps (in the style of Pierre Novat); namely, **cast shadows** or simply put, it aims to analyze different components of cast shadows on mountainous terrain such as changes in hue due to aerial perspective, abstraction of shape and overall changes in intensity along a slope. Additionally, artistic nuances have been considered by analyzing a wide variety of panoramas (around 60 panoramas spanning nearly 50 years) and through interviews with the artist, Arthur Novat.

1.2 Research questions & motivation

Cast shadows on terrain form one of the most important components in artistic depiction of landscapes. Specifically, in the context of panorama maps, subtle changes to hue or intensity in different segments of these shadows can effectively change user perception. In fact, by varying these *artistic-conditions*, spotlighted segments can have a different meaning to multiple user-types from experts to casual level skiers, map surveyors, hikers etc. Therefore, it becomes imperative to analyze and understand these artist-defined conditions to not only allow for flexibility in map-creation but also for greater customization without professional/artistic intervention.

Along these lines, several research questions can be articulated :

- How do cast shadows portrayed in panorama ski-maps interplay with the **shading** used by the artist?
- How do structural properties of cast shadows (**shape**, **intensity** and **hue**) influence visual perception in ski-panorama maps?
- How does **inter/intra - artist style** affect depiction of cast shadows in ski-panorama maps?

Before proceeding, it is important to understand the motivation behind pursuing such a cross-disciplinary topic. The following points give a brief insight :

- Ski-maps that use rendered terrain to depict certain landscapes look comparatively worse when compared to the ones that are artistically driven.(consider Figure 1.4 as an example).
- With artist-drawn maps, the shading, shadows, texture details as well as terrain geometry can be controlled and modified to incite the observer's focus to key areas on the map.
- Time and cost are also factors to be considered when maps are being commissioned. The process (artist survey to finalization) nearly takes 6-8 months of work and is fairly expensive, which in turn leads ski-stations to pursue cheaper alternatives. Automated techniques to replicate artist style could help in the resurgence of artist-drawn ski-panorama maps.
- The style of art is fairly niche and requires a high degree of specialization in cartography which results in apprentices being few and far in between. With the effort required to learn both the artistic style and modeling terrain (to fit a specific narrative) and a general lack of guidance, sustaining a panorama-driven art-studio is difficult.

1.3 Main contributions

With the research questions and motivational factors in mind, the goal of this thesis is to create a framework that best describes cast shadows and it's structural properties along with a stylized shading model, in the style of atelier Novat. To break it down further, the contributions can be listed as follows:

- A rendering framework that aims to convert a batch of DEM files into an image reciprocating the style of Atelier Novat.
- A novel, filter-based, stylized shading method that accentuates cast shadows and it's corresponding properties while following the conditions set by the artist.
- Set of image-space methods (implemented on the GPU via shaders) defining conditions for abstraction of shadow shape, changing intensity levels along mountain slopes and differences in hue from the viewpoint to the horizon. Some of the methods involved, mimic (to an extent) physically-based approaches, forming an interesting contrast between art and realism.
- A brief study on prominent ski-panorama art-styles prevalent in Europe and North-America.

In the following chapters, the thesis will survey background literature, provide an introduction to the rendering framework including methods used and implementation, discuss results and their shortcomings and finally wraps it up with the conclusion and future work as an ending note.

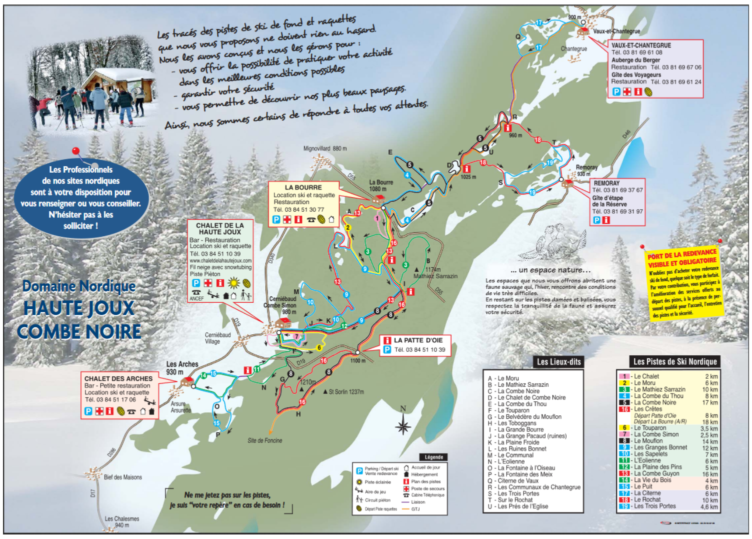

Les tracés des pistes de ski de fond et raquettes
que nous vous proposons ne doivent rien au hasard.
Nous les avons conçus et nous les gérons pour :
- vous offrir la possibilité de pratiquer votre activité
dans les meilleures conditions possibles
- garantir votre sécurité
- vous permettre de découvrir nos plus beaux paysages.
Ainsi, nous sommes certains de répondre à toutes vos attentes.
Les Professionnels de nos sites nordiques sont à votre disposition pour vous renseigner ou vous conseiller. N'hésiter pas à les solliciter !
Domaine Nordique
HAUTE JOUX
COMBE NOIRE
LA BOURRE
Location ski et raquette
Restauration
Tél. 03 84 51 30 77
CHALET DE LA HAUTE JOUX
Bar - Restauration
Location ski et raquette
Tél. 03 84 51 10 39
www.chaletdelahautejoux.com
Fil neige avec snowtubing
Piste Pièton
LA PATTE D'OIE
Tél. 03 84 51 10 39
CHALET DES ARCHES
Mignovillard 880 m
La Bourre 1080 m
Cerniébaud Combe Simon 980 m
Cerniébaud Village
Mathiez Sarrazin
Les Arches 930 m
Vaux-et-Chantegrue
Remoray
Les Lieux-dits
Les Pistes de Ski Nordique
Ne me jetez pas sur les pistes, je suis "votre repère" en cas de besoin !

Figure 1.3: A 2-D ski-map with routes splatted on a rough shape of the landscape

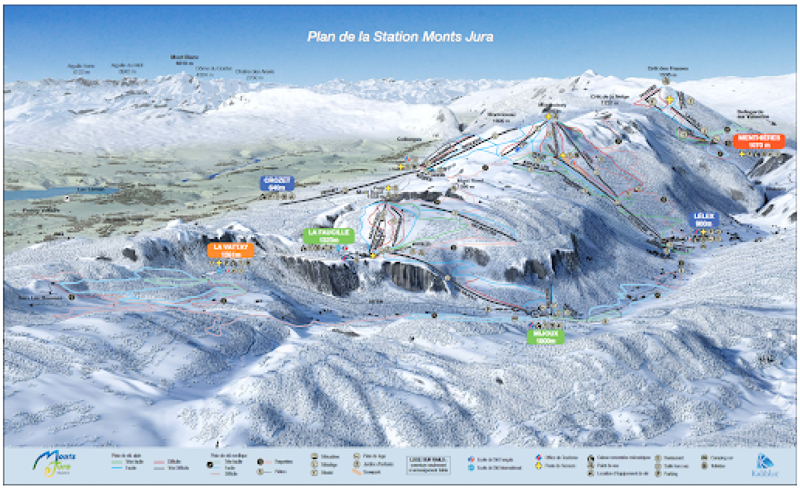


Figure 1.4: A ski-map of mount Jura with 3-D terrain created from high-resolution aerial-imagery (Kaliblue©)

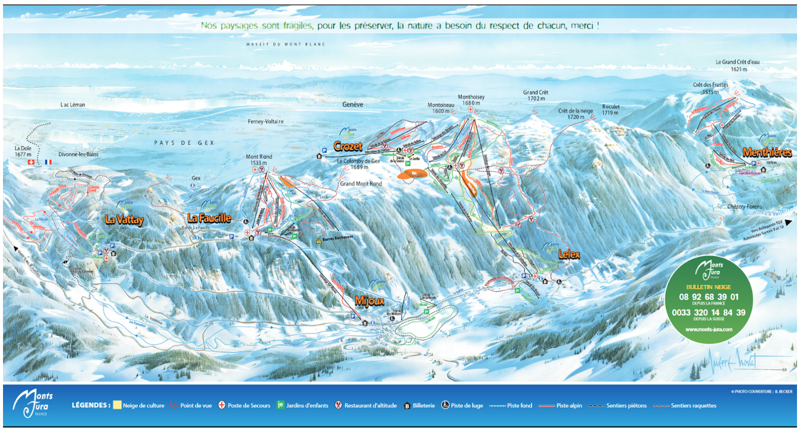


Figure 1.5: A ski-map of mount Jura drawn by Pierre Novat. (Notice the changes in Hue, viewpoint and terrain)

2 Background

This chapter is mainly divided into three areas of focus; a theoretical background on some of the general components used in the proposed rendering framework, a brief overview on artistic differences in panorama maps and their relevance towards the theme of this thesis. Finally, a short section on recent work done on panorama map rendering over the past decade to conclude the literature review.

2.1 Rendering of terrain

Terrain rendering has been a popular topic in computer graphics for quite a number of years, which is why it is increasingly used in domains such as virtual reality, GIS, flight simulations and landscape editors. In order to truly represent a piece of terrain digitally, an efficient mapping criterion must be used and proper real-world measurements must be undertaken. This section expands on two topics, acquisition and exploration of topographical data followed by a brief introduction to generation of height-maps. Moreover, this gives a bit of insight into the initial input given to the rendering framework (detailed in Chapter 3)

2.1.1 Acquisition and mapping of topographical data

The problem of terrain data acquisition is a fairly old one; before the usage of DEMs (Digital Elevation Models), a format known as DTM (Digital Terrain Model) was created and used. In fact, a *Digital Terrain Model* was defined by [Mil57; JD78] as an ordered array of numbers that represents the spatial distribution of terrain characteristics, which appears on the XY-coordinate system with terrain variation being a modulation of it's Z-ordinate. It's inception was mainly to hasten the design of highways by digitally computing photogrammatically acquired terrain data.

The process of photogrammetry and remote sensing was fairly complex and tedious in those times. An instrument known as the *Kelsh plotter* (also known as a stereoplotter, shown in Figure 2.1) determined terrain elevation levels from satellite-captured stereo photographs. Although it was useful in function, the amount of precision required to operate it was really high. However, with time, these plotters were replaced by modern acquisition tools that were considerably less dependent on human error. A popular example is LIDAR (light detection and Ranging) which is actively used in remote sensing and surveying operations. Its main advantage lies in the fact that the output is in the form of three-dimensional points in object space which in turn, helps in raw DEM synthesis.

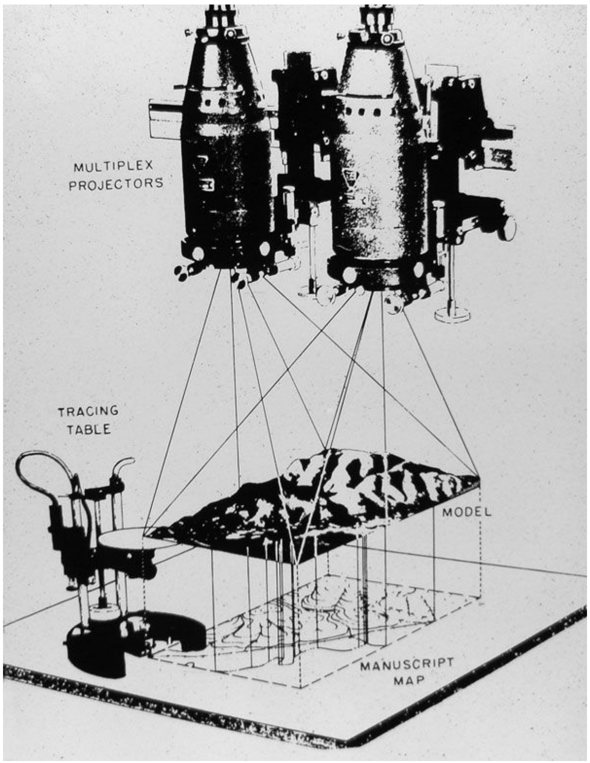


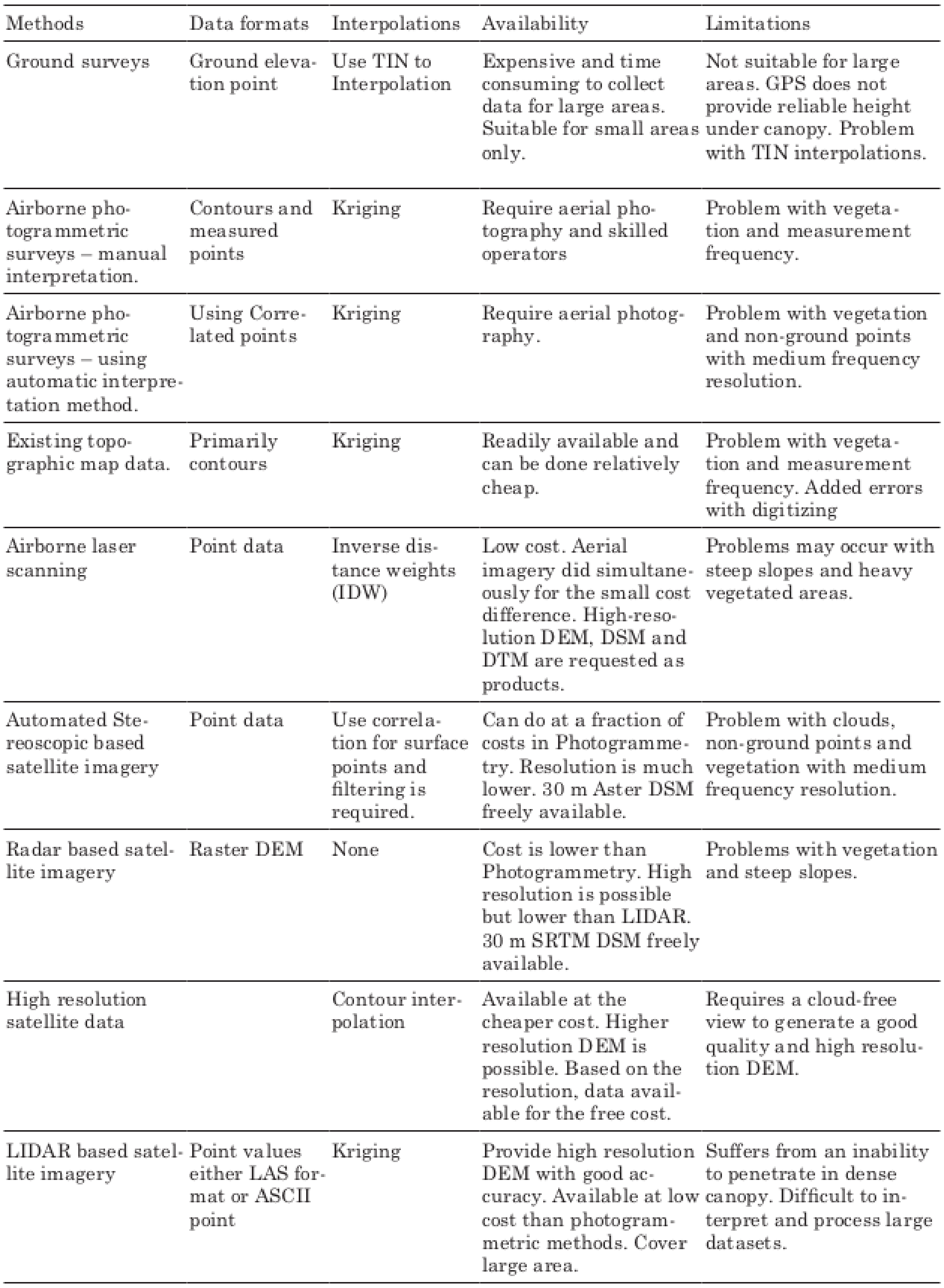
Figure 2.1: Schematic representation of a multiplex stereoplotter, USGS

In terms of reliability, LIDAR provides higher accuracy and density when compared with regular photogrammetry methods. It also helps in the creation of forested area DEMs with an accuracy rivaling that of open areas captured by photogrammetry methods [KP98]. Other notable advantages include shadow-free data and accurate urban area generation. However, not all data is created equally; LIDAR-based captures suffer from inherent noise during airborne scans. Additionally, the captured topography may contain traces of vegetation, humans and animals as the light tends to echo off of non-ground points (should the scene contain any). Therefore, in order to get DEM data that is cleaner (relatively noise-free) and more reliable, filtering of LIDAR-based DEM formats must be done.

DEMs are generally present in multiple formats; in terms of shape, we can divide them into three different types : regular-rectangular grid, Triangular Irregular Network (TIN) and the contour line model [Ram06]. Out of these three, the regular grid is the simplest and most efficient as the packing structure is similar to the array data structure in computers [Ram06; Zia07]. Furthermore, the density and sampling rate is increased through interpolation methods [Liu08], e.g. deterministic and geo-statistical interpolation. This is done in order to create a highly-dense, packed grid-like format.

Another important parameter that is considered when evaluating DEMs is *DEM resolution*. It can be defined as the total grid size of the DEM expressed in ground distance. The general idea is to obtain a grid that contains the an adequate description of the terrain with the least amount of data or defined by the requirements of the application [Gao97]. Still, DEM files contain more detail than the requirement for the process being modeled [Zia07].

To summarize the section on terrain data acquisition and DEMs, the Figure 2.2[1] below gives a brief description on the advantages and challenges based on formats, interpolators, limitations and availability of DEM sources.



Methods	Data formats	Interpolations	Availability	Limitations
Ground surveys	Ground elevation point	Use TIN to Interpolation	Expensive and time consuming to collect data for large areas. Suitable for small areas only.	Not suitable for large areas. GPS does not provide reliable height under canopy. Problem with TIN interpolations.
Airborne photogrammetric surveys – manual interpretation.	Contours and measured points	Kriging	Require aerial photography and skilled operators	Problem with vegetation and measurement frequency.
Airborne photogrammetric surveys – using automatic interpretation method.	Using Correlated points	Kriging	Require aerial photography.	Problem with vegetation and non-ground points with medium frequency resolution.
Existing topographic map data.	Primarily contours	Kriging	Readily available and can be done relatively cheap.	Problem with vegetation and measurement frequency. Added errors with digitizing
Airborne laser scanning	Point data	Inverse distance weights (IDW)	Low cost. Aerial imagery did simultaneously for the small cost difference. High-resolution DEM, DSM and DTM are requested as products.	Problems may occur with steep slopes and heavy vegetated areas.
Automated Stereoscopic based satellite imagery	Point data	Use correlation for surface points and filtering is required.	Can do at a fraction of costs in Photogrammetry. Resolution is much lower. 30 m Aster DSM freely available.	Problem with clouds, non-ground points and vegetation with medium frequency resolution.
Radar based satellite imagery	Raster DEM	None	Cost is lower than Photogrammetry. High resolution is possible but lower than LIDAR. 30 m SRTM DSM freely available.	Problems with vegetation and steep slopes.
High resolution satellite data		Contour interpolation	Available at the cheaper cost. Higher resolution DEM is possible. Based on the resolution, data available for the free cost.	Requires a cloud-free view to generate a good quality and high resolution DEM.
LIDAR based satellite imagery	Point values either LAS format or ASCII point	Kriging	Provide high resolution DEM with good accuracy. Available at low cost than photogrammetric methods. Cover large area.	Suffers from an inability to penetrate in dense canopy. Difficult to interpret and process large datasets.

Figure 2.2: Source : Caribbean Handbook on Risk Information Management

[1] http://www.charim.net/use/92

2.1.2 Height-map texture generation

In the previous subsection, acquisition of real-world terrain data and it's corresponding distribution as DEMs was discussed. Procedural modeling of terrain is a topic that has seen active research over several decades. The main gist behind procedural generation is the fact that instead of having to manually create models from scratch, a procedure is designed that allows for automatic content creation.

In the rendering framework (discussed in the next chapter), most of the techniques are applied with a height-map texture as input. Height maps are basically two-dimensional grids of elevation values represented in the form of a grayscale texture or image. Nevertheless, from the previous section, it can be noted that not all DEM files are in perfect condition; some suffer from the lack of data, irregular mapping of heights due to natural or artificial occluders (vegetation, animals and so on). Additionally, if the sole purpose is to demonstrate terrain rendered in the form of a specific art-style rather than ski-map creation or if there are specific applications of said art-style in games/ visual-effects etc., procedural generation of terrain can be a viable alternative.

In terms of creation, some of the oldest techniques employ the use of sequential subdivision. For example, multiple variants of the mid-point displacement method [Mil86] were used wherein, an elevation point's value is taken as the mid-point between it's neighboring elevations plus a random offset. The offset in this case serves as a controllable parameter based on the roughness of the height-map texture. Nowadays however, a lot of the height-maps that are generated rely on multiple variants of noise (strictly following the domain of signal processing) [Per85], specifically the use of fractals. In fact, in order to obtain naturally appearing crests and troughs on the terrain, multiple levels of modulated and re-scaled noise is stacked on top of one another.

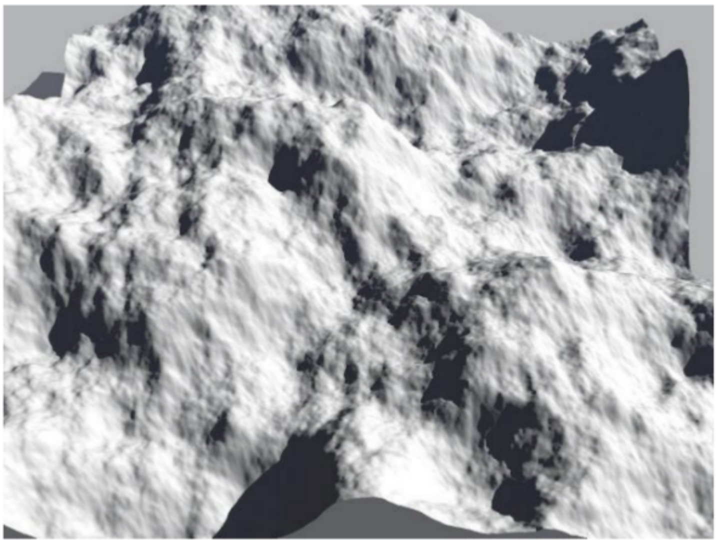


Figure 2.3: An ordinary fractal Brownian Motion (fBM) terrain patch of arbitrary dimensions [Mus]

An important example of noise that is frequently used to create realistic terrain is *fractal Brownian Motion* (fBM) noise. Fractals are a mathematical concept that is capable of producing multiple, varied patterns. In fact, it can rightly be defined as "a geometrically complex object of which arises through the repetition of some shape over a range of scales" [Mus]. Fractals can fundamentally be seen in nature in the form of clouds, mountains, trees, turbulence etc.

The implementation of such a fractal height-map texture is a relatively simple process [Mus]. Depending on the style of the terrain needed, multiple controllable parameters such as lacunarity (frequency of each iterative step), octaves, bandwidth etc. can be used to tweak the terrain into its required shape. Normally, basis functions such as Perlin, Worley or Voronoi functions are used in the creation of fractal height-map textures.

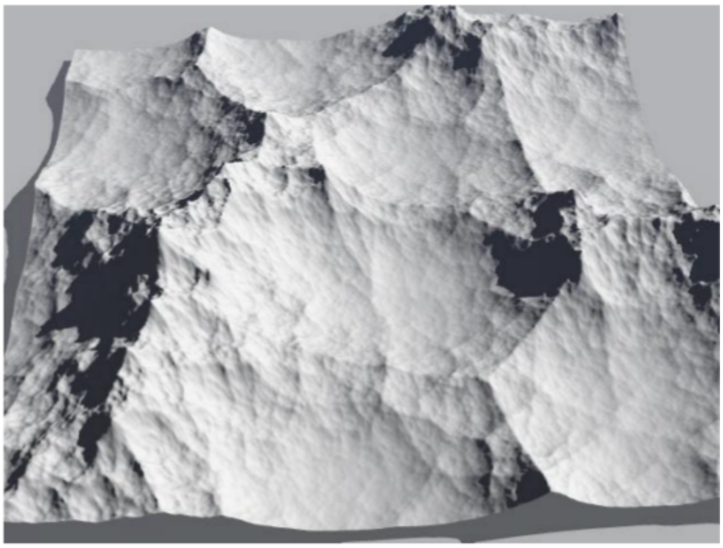


Figure 2.4: The Mordor terrain model : a hybrid multi-fractal model made from combining a Worley-Voronoi distance-squared basis [Mus].

2.2 Non-Photorealistic Rendering

Non-Photorealistic Rendering (NPR) is referred to as the branch of computer graphics that focuses on the representation of artistic styles and it's application to images or three-dimensional scenes. In the context of this thesis, it plays a central role within the proposed rendering pipeline; techniques that are used in the stylization aspect of the pipeline mainly draw their inspiration from the methods described in this section. In fact, most of the techniques are implemented in image-space and are filter-based approaches. This not only makes it easier to test and prototype but also eases the depiction of scenes.

The following subsections will focus on three areas; watercolor-based effects, opaque-paint based effects (with an emphasis on gouache and oil) and artistic stylization of shadows.

2.2.1 Watercolor-based stylization

Watercolor or *aquarelle* is a painting method that uses colored, non-water soluble pigments held in suspension within a liquid medium as a dispersion in order to create art. Semantically, watercolor refers to both, the translucent medium as well as the artwork created by it. Although the focus will mainly be on the artwork side of things, it is important to understand the chemical composition of these paints as it helps understand the motivation behind some of the well-known watercolor effects (granulation, edge-darkening etc.).

Most of the paint manufacturers have a pre-defined quality standard when it comes to tubes of watercolor paint. Each tube is separated into two sections (without physical dividers but more-so as a dispersion); **color**, composed of the pigment and brightener, and the **vehicle**, composed of several supporting chemical compounds such as binder, plasticizer, humectant, fillers and additives (as shown in Figure 2.5).

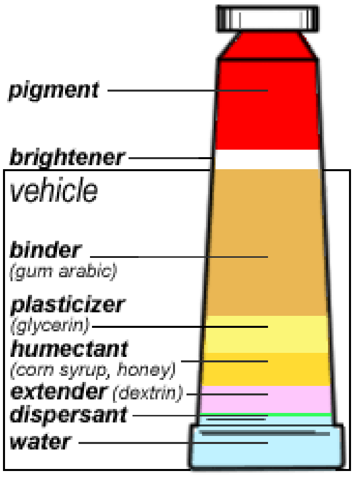


Figure 2.5: Schematic backbone composition of a modern watercolor paint

In terms of tone control (shade or tint), the brightener present along with the pigment, plays an important role in lightening the chroma of dried paint. Additionally, paint color can be changed depending on a variety of factors such as percentage composition of pigments, dispersability, tinting strength etc. The "vehicle" (as shown in Figure 2.5) consists of several compounds that are designed to preserve the appearance of the pigment, once it has been applied onto a surface. Depending on the extent of usage of these compounds, a variety of effects can be observed which are fundamentally recognized as *watercolor effects*.

Watercolor as an art-form is extremely old, dating back to the paleolithic era in the form of cave paintings. Throughout history, there have been surges in watercolor usage as a medium of art to paint botanical, landscape and wildlife pieces. Europe (mainly during the middle ages and the Renaissance period) was specifically involved in the propagation and distribution of watercolor, only to be replaced by the more opaque paint, Gouache. It was, however, picked up by prolific painters in the United States (circa 19th century) and popularized as the "new-world" art.

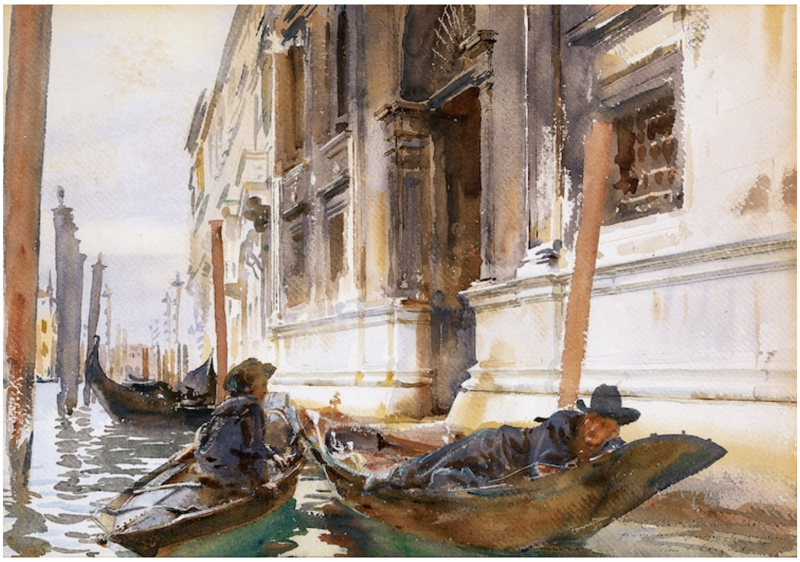


Figure 2.6: Rise in watercolor (19^{th} - 20^{th} century) : "Gondoliers' Siesta", John Singer Sargent, 1904

In terms of the paint itself, watercolor is widely popular because of three reasons; **abstraction**, **translucency** and **layering effects**. In the hands of a competent artist, watercolor can be quite expressive. The ability of colors to showcase translucency and opacity (see Figure 2.7) is nigh unmatched. In fact, multiple colors can be blended on top of one another in the form of layers to create vibrant color palettes within the painting. An interesting characteristic of watercolor lies in the lack of detail present in the paintings; since watercolor mainly relies on water as a medium of dispersion for the pigment to set in, controlling and setting definite boundaries for shapes becomes increasingly difficult. This results in paintings that appear highly abstract but have a flow between each section of the painting.

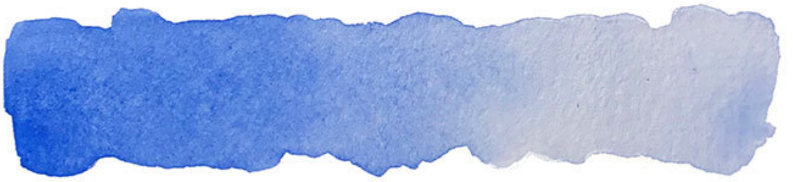


Figure 2.7: Changes in pigment tone with water dilution showcasing translucency

Given the rich history of watercolor, it was only natural that it became a topic of interest in the realm of computer graphics. The idea of representing an image in the form of watercolor and allowing artistic control over images, is something that has been undergoing significant research for the past two decades. Over the years, there have been a number of methods that focus on replication of watercolor-based art; most significantly, many approaches can be broadly categorized into two forms: physical simulation of watercolor and replication of watercolor purely as a visual effect (either in image space, through filter-based methods or object-space methods).

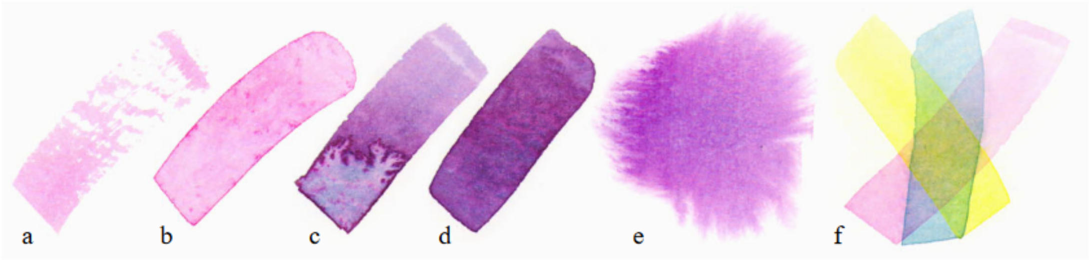


Figure 2.8: Watercolor effects: drybrush (a), edge darkening (b), backruns (c), granulation (d), flow effects (e) and glazing (f) [Cur+97]

The thesis will not delve too deep into either of these two forms but will try to elaborate on the effects that can be physically simulated. The methods described in the proposed rendering pipeline fall into the latter of the two forms. In terms of watercolor simulation, the seminal work by [Cur+97] continues to inspire modern-day approaches. A brief description of the effects described in [Cur+97] can be given as follows: (supporting Figure 2.8)

- **Dry brush** : A rough, uneven trail of pigment when a dry brush has contact with raised portions of the paper at a grazing angle.
- **Edge Darkening** : A wet brush on dry paper technique that results in the outward migration of pigments to the edges.
- **Backruns** : As the name suggests, when the water tends to collect into a damp region of paint, the pigments get pushed outward in a branched manner resulting in darkened edges.
- **Granulation** : A grainy texture emphasizing crests and troughs on the watercolor paper. It is heavily dependent on the type of pigment as well as the water deposition on the paper.

- **Flow** : An example of a wet-on-wet technique (brush-paper interaction) that allows for a continuous, diffused color gradient. The shape of the blob produced generally follows the direction of water.
- **Glazing** : It is an effect that is produced when watercolor pigments are layered one on top of another. Due to the properties described earlier on in the subsection, this produces a translucent, even effect. This forms one of the most important effects of watercolor.

Now that we have described (in brief) the main effects of watercolor, it is important to understand that representing these effects through physical simulation is not always desirable. The paper by [Cur+97]. goes in depth into each specific effect and gives significant emphasis to particle (as pigments) interactions. In fact, there are several disadvantages to this specific approach; the main one being that physically simulating the behavior of particles is computationally expensive, can be slow and is adequate for real-time purposes. Additionally, color compositing differs from a screen to real-life scenarios. For example, combining multiple pigments through software may result in a completely different color than the one that is expected.

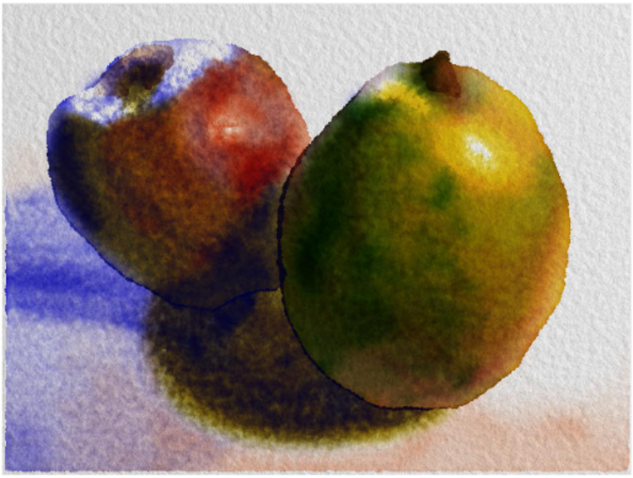


Figure 2.9: Watercolorization of an image as rendered with the method proposed by [Cur+97]

Given these challenges, alternative representations that focus on imitation rather than simulation, can be considered. Recently, over the past decade, image-space methods for replicating watercolor as a post-processing effect have been gaining traction. In the paper by [Bou+06], a framework is proposed that takes an image or 3-D scene and converts it into a watercolor-like image through the use of various filters. The watercolor-effects that described prior are replicated through image-space filtering of the input scene/ image and do not in any way modify the original input. Additionally, most of the effects are reproduced using several 2-D textures in the pipeline. For example, for effects like wobbling, pigment dispersion and turbulence flow, a series of 2-D noise textures are used to simulate the pseudo-random movement of the watercolor pigments.

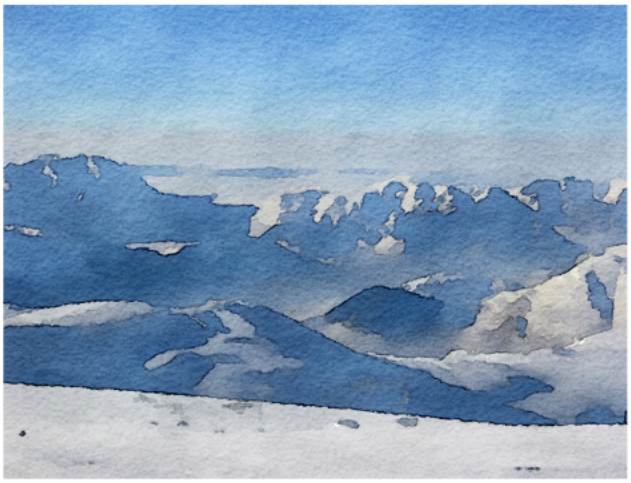


Figure 2.10: Watercolorization of a photograph as rendered with the method proposed by [Bou+06]

In this thesis, the proposed rendering framework aims to take inspiration from some of the methods such as edge-darkening, region-based abstraction and tone-mapping, as described by [Bou+06]. Additionally, detailing these methods not only helps in understanding the proposed stylized shading approach but also, the artist's intentions when painting a specific section (shadow or lit region) of the panorama map.

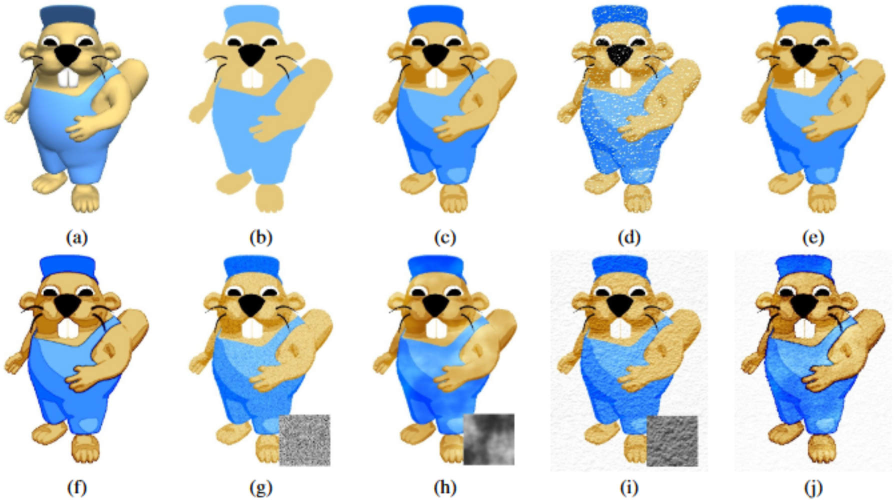


Figure 2.11: Static pipeline illustrated for a 3-D model : (a) 3-D model (b) Original color chosen by user (c) toon shading (d) dry brush (not used in this example) (e) wobbling (f) edge darkening (g) pigment dispersion layer (h) turbulence flow layer (i) paper texture (j) final result [Bou+06]

2.2.2 Opaque paint-based stylization

Unlike watercolors, this subsection will not go in-depth into paint consistency or composition but rather its visible applications within the field of computer graphics. As the name suggests, opaque paints generally do not allow light to bounce through them, resulting in colors and variants of said colors to be solid, precise and more defined (in terms of shapes). Paints are said to be opaque based on a combination of factors such as pigment percentage, layering ability and color blending. Additionally, there are three different types of paints that are have been used frequently by artists over centuries:

- **Gouache and Tempera** : Matte finish when dried and is also known as opaque watercolor. Can be re-wetted and is mainly used commercially in posters, comics etc.
- **Oil** : Slow-drying with a high viscosity (can be modified by mixing varnish or turpentine, may result in a change in glossiness). Originally it was not used as an art form but rather as a preservative for wooden and metal structures.
- **Acrylic** : Fast-drying paint with the pigment suspended in a polymer emulsion. Most flexible medium out of the three and can be made to resemble watercolor, gouache or oil based on dilution with water, gels or other mediums.

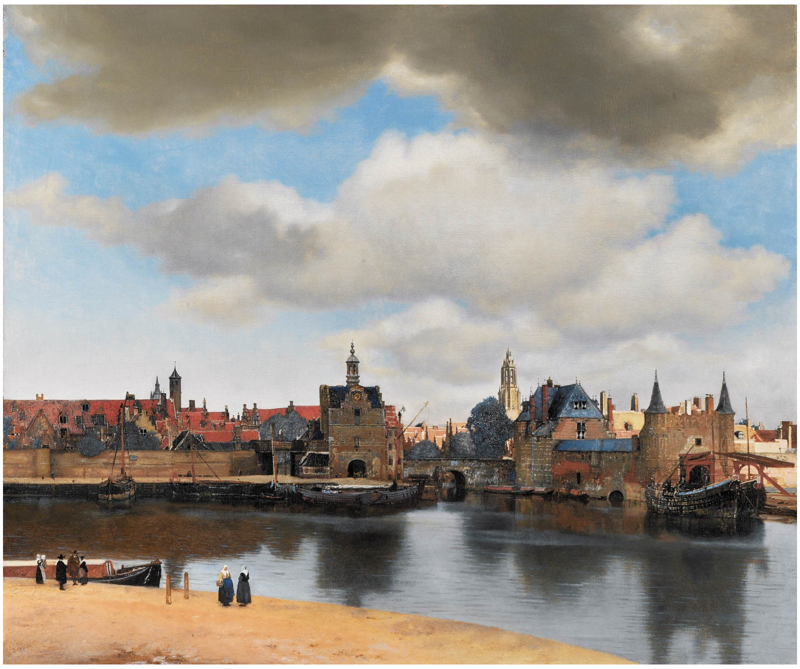


Figure 2.12: "View of Delft", Johannes Vermeer, Oil, 1660

This subsection will focus on the first two paints (gouache and oil) and subsequently, look into computational methods used in replicating artistic effects observed in oil paintings. Gouache (from the Italian word, *Aguazzo* for "mud") was mainly used by landscape painters in the 18th century and is also probably the most controversial technique in watercolor painting history. Initially, it was used purely for decorative purposes in manuscripts and tapestries by French and English artists. Later on, due to it's opaque nature and the ability to represent near-solid colors, it was popularized in landscape paintings alongside watercolor usage.

In terms of composition, gouache has a few distinct characteristics that set it apart from other paints (specifically watercolor), these are:

- Much thicker paint layer, allowing for greater coverage on painting paper without much support or difference to finished color appearance.
- Cannot be diluted with water to lighten the color (in comparison to watercolor) but needs to mixed with a white pigment (similar to oil).
- Glazing and tinting is not possible as the layers are opaque and far too thick for light to permeate through.
- Due to it's chemical composition, paint stays on the surface of the paper, rather than being absorbed. This allows for greater control in texture variations as opposed to watercolor (limited to pigment-medium diffusion effects).
- Finally, gouache allows for flatter color areas as the pigment-filler mixture is heavier and does not diffuse readily. This allows for a vibrant color range in paintings.

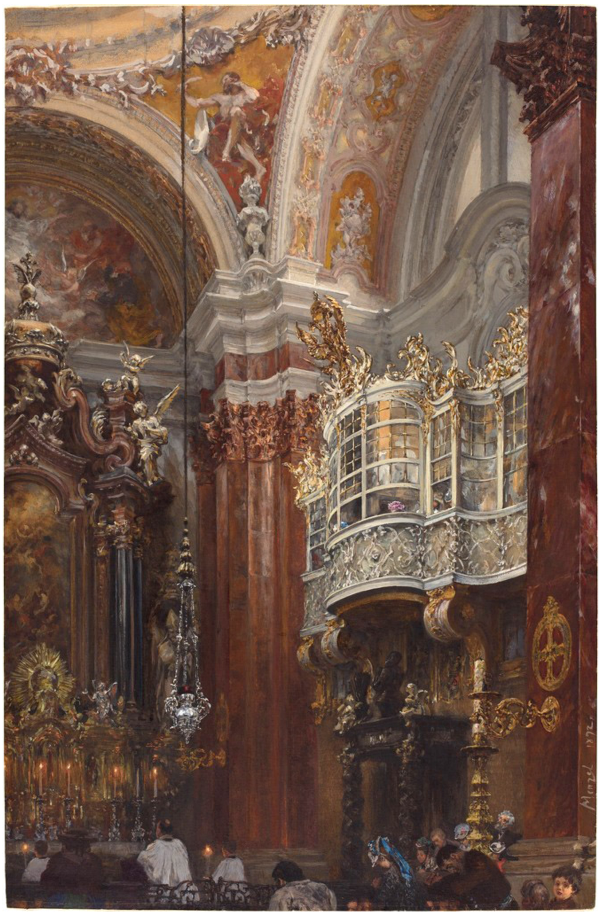


Figure 2.13: "The Interior of the Jacobskirche at Innsbruck", Adolph Menzel, Gouache, 1872

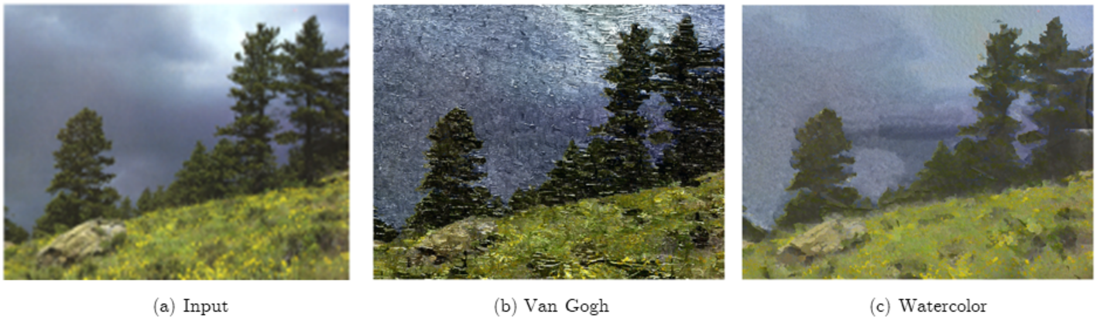
To reiterate, gouache is used for illustration and photo-reproduction due it's brilliant color range (especially in the red, magenta and violet spectrum, visible in figure 2.13). However, it's similarities with oil-based paints also serves as it's disadvantage; depending on the thickness of gouache layers applied, the paint can crack and discolor. Additionally, when used in conjunction with watercolors, gouache is mainly applied as highlights or to accent forms and clarify details. In the context of panorama maps painted by Novat, gouache (and at times, acrylic) is used for streaks of snow highlights in certain regions of the painting (to create a raised, "popping" effect).

Apart from gouache, the artistic effects present in oil paintings have been studied extensively in the computer graphics. Usually, the effects observed are a combination of stroke-based techniques that appear to simulate the effect of paint-on-canvas through soft color blending and no visible, sharp edges. More recently however, there has been significant progress in the usage of image-filtering for stylization (shape-adaptive smoothing [EKD08], anisotropic diffusion filtering [Wei] and shock filtering [KL08].As our work focuses on image-space methods, object-space, stroke-based operators will be omitted from this section.

There are however, issues when representing an oil-painting purely through image filters, these problems arise in the form of:

- Optimization of the global color distribution in order to preserve contrast levels of important features.
- Orientation of the paint textures to reflect the artist's impressions.
- Most importantly, the level of control that must be provided for creating local edits.

With the focus being on image-space stylization, the approaches can be categorized in three ways : example-based texture transfer, region-based feature segmentation and edge-preserving smoothing techniques. Example-based texture transfer, as the name states, refers to a set of methods that rely on transferring textual features based on an example/ source image (can be an artwork or photograph). There are a couple of approaches [Her+01] that discuss the procedure of mimicking painting styles through image-analogy based texture transfer and the use of machine learning to "learn" the artistic style.



(a) Input (b) Van Gogh (c) Watercolor

Figure 2.14: Paintings by example [Her+01] (a) Input image (b) Van Gogh style transfer (c) Watercolor style transfer

Region-based feature segmentation methods are a bit of an oddity, in the sense that, the images that are produced as a result of such algorithms are highly abstract in nature; a property that is rare in oil paintings. However, certain methods [DS02] allow the usage of image abstraction as an intermediate step to achieving oil-painting-like consistency in images. Other use-cases for such methods include texture, style replication of stained glass or fabric-art.

Finally, in order to modify local features to a high degree or to retain other portions of the image without extensive abstraction, edge-preserving smoothing methods can be used. A few examples include the bilateral filter [PD06], guided filter [HST10], kuwahara filter [KKD09] and mean-shift segmentation [CM02]; the first three are used and explained in the following chapters. Additionally, image-operators such as morphological erosion and dilation along with gradient minimization [Xu+11] can be used to simulate a painterly effect.

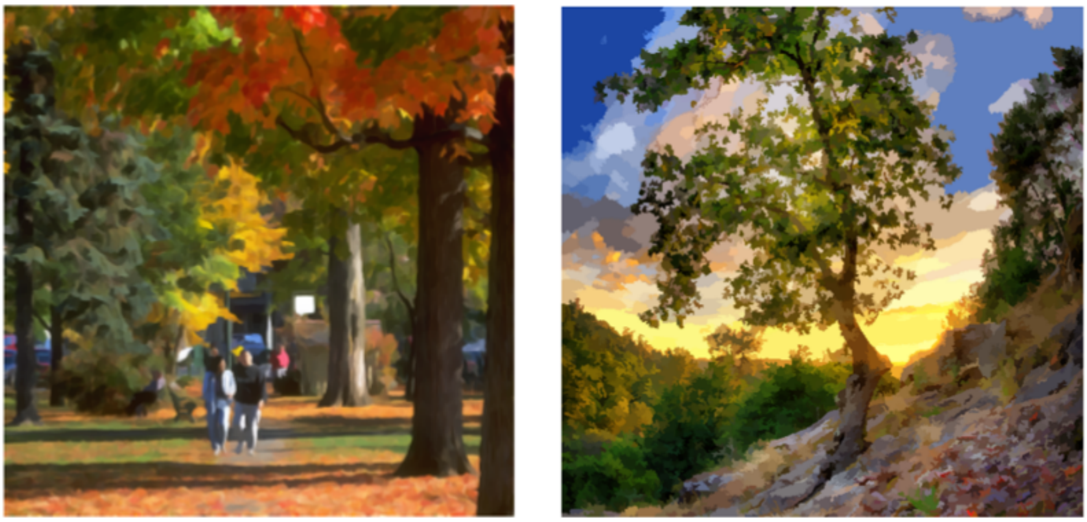


Figure 2.15: Anisotropic kuwahara filter [KKD09] (left) and mean shift segmentation [CM02] (right)

2.2.3 Artistic stylization of shadows

Before describing the methods for stylizing shadows, properly defining the term "shadow" would prove helpful in understanding it's properties as well as the need for stylization. Shadows have multiple definitions depending on the context and the field of study that it is used in. For the purpose of this thesis, a *shadow* (in computer graphics terms) can be defined as [Has+03] : "the region of space for which at least one point of the light source is occluded". There are two important assumptions considered here [Eis+11], first, the illumination is assumed to be direct illumination and light bounced off of other surfaces is ignored. Secondly, the occluders in the path of light are considered to be opaque (an assumption that may not hold true in real world scenarios).

In the context of panorama maps drawn by Novat, there are several points that need to be considered that may not align with the assumptions stated above:

- Heavy emphasis on the usage of hard shadows (crisp, well defined edges).
- Shadows have abstract, rounded boundaries that aim to blend with valley crevices.
- Shadows have gradients within them which may not conform to the idea of them being considered pure binary visibility maps.
- Heavy emphasis on reflected light from surrounding mountain faces present within the interior regions of shadows.
- Hue changes depending on viewer position and the position of the shadow itself within the panorama painting.

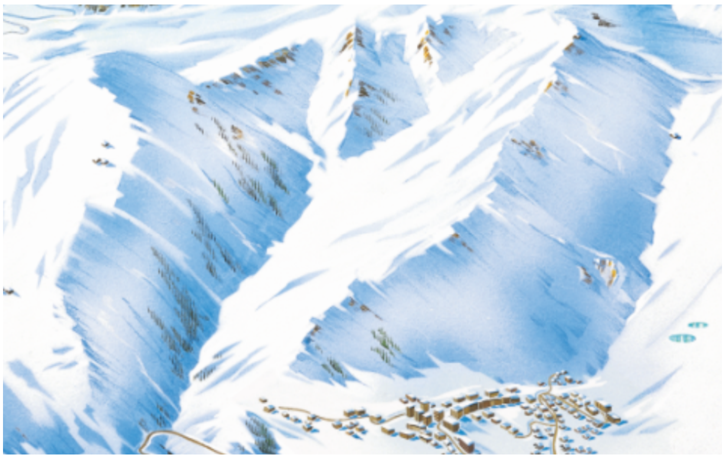


Figure 2.16: Cropped from a painting of Sybelles by Pierre Novat, showcasing the variation in cast shadows

From Figure 2.16, it can be noted that human perception of the scene can be changed by changing certain properties of shadows. In cinematographic production, lighting and compositing artists typically work hand-in-hand to ensure that a scene is represented with proper artistic direction. In most cases modifying certain properties of shadows (size, boundary, shape etc.) allows for better creative control. This subsection aims to introduce a few methods in stylizing shadows.

In the seminal work by [DeC+07], four parameters are introduced for controlling the visual quality of shadows (shown in Figure 2.17), these are:

- **Inflation** : A parameter that allows controlling the size of shadows. Increasing the parameter value gives a larger version of the shadow emanating from the object.
- **Brightness** : Controllable intensity of shadow when fully occluded or when exposed to multiple indirect sources of illumination.
- **Softness** : Shift in width of transition region from fully occluded to fully visible, simulating the effect of an area light.
- **Abstraction** : Controlling the shape of shadow boundaries. Lower values of the controllable parameter retain original shape, higher values result in rounder, simplified shadows.

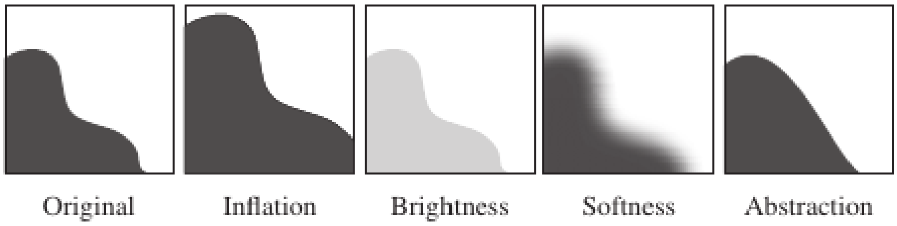


Figure 2.17: Paramters introduced for controlling visual quality of shadows [DeC+07]

Within the approach proposed by this thesis, abstraction, brightness and softness are emphasized. Additionally, these parameters are influenced in image-space with controllable parameters (for filters used to change shadow properties). Another interpretation of modifying shadows involves the use of a click-and-drag interface [Rit+10] which allows for local editing and deforming of shadows. Although this allows for far greater creative control, the proposed approach by this thesis would strictly focus on global, parameter-based editing of shadows in order to further automate the process of stylization. However, methods proposed by [MIW13] and [DVP19] in the same vein open up interesting avenues for future work.

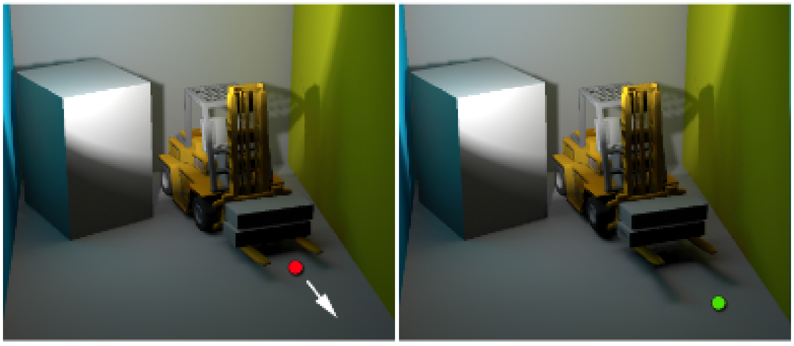


Figure 2.18: Shadow deformation by a click-and-drag interface [Rit+10]

2.3 Atelier Novat : An overview

This section is mainly intended to give the reader a brief overview of the overall painting process and artistic intricacies of the "Atelier-Novat" art-style. For the past sixty odd years, Pierre Novat was the artist responsible for painting scenic, landscape panoramic views of the ski-areas in alpine France. In order to produce ski-maps that were legible and interpretable by skiers, Novat worked with local sponsors such as ski-resorts and cartographers. With time, as his popularity increased, he received commissions for summertime panorama maps as well as maps extending into alpine Swiss territory.

With the passing of Pierre, his mantle has been inherited by his son, Arthur Novat. As the process and artwork is fairly niche, Arthur has been trying to share his knowledge and experience with fellow apprentices as well as researchers interested in preserving cultural heritage. Preservation of this artstyle along with automation of said artistic techniques forms one of the most important motivational factors to pursue research in this direction. From 2015 - 2017, a collaboration between cognitive geographers (INRIA), historians (LAHARA) and computer scientists (LIG) was established under the MECOMO (MÉmoire, COnnaissance et MOdélisation de la Montagne) project. The objective being : to reconstruct a chronological representation of panoramas by Novat from the 1960s and better understand the artistic guidelines that drive the creation of such panorama maps. Although, there have been several studies published, there is still scope for research in the direction of automatic creative control and rendering of such maps.

Additionally, for the scope this thesis, the findings are based on empirical data in the form of Pierre Novat's book ("Plans des Pistes", containing a large collection of drawn panoramas from the '60s to the early 2000's), interview with Arthur Novat and prior research notes collected from Frédérique Novat, Arthur Novat and Laurent Belluard. It is to be noted that even though certain assumptions (regarding shadows, shading etc.) may hold across a large variety of panoramas, there might be certain outliers that do not conform.

2.3.1 Artistic constraints

There are a number of problems that might occur during the artistic process of painting a panorama map. To begin with, as stated before, commissions are usually received in the form of local ski-resorts, authorities or sport committees such as organizers for the winter Olympic games. Usually, the outcome for a panorama map is never photo-realistic; in the sense that, the sponsor desires a non-realistic viewpoint of a specified mountain or area. For example, if the ski-resort resides in a mountain cavity that is usually not visible. It is upto the artist to *artificially* emphasize this resort and it's surrounding environment either by deformation or through visual, artistic cues.

The sponsor has a considerable amount of influence on the resulting panorama, which may include complete deformation, route formation and hue highlights specific to the location chosen by the sponsor (usually a specified mountain slope or valley) Additionally, there are a few conditions that a "panorama ski-map" should follow. These are set in order to let the skier better acquaint themselves with the terrain and identify potential landmarks. The constraints are given as follows :

- Optimal weather and lighting conditions. For example, clear skies with a low morning sun revealing the entire terrain.
- Height information should relatively stay the same. Modulating heights of peaks or valleys drastically is forbidden.
- Gradient levels must be respected as well; steeper the slope, steeper the appearance in the panorama map.
- Balance between sponsor and local populace map renditions; regular skiers and locals should be able to recognize the terrain without having to compromise for an non-realistic view of the sponsor's ski-resort or mountain.
- Ski-track levels must be maintained (green, blue, red or black). Viewer should be able to distinguish between green and black tracks by looking at the shape of slopes (straight and steep for black, relaxed and curved for green)

2.3.2 Painting process : stages & pipeline

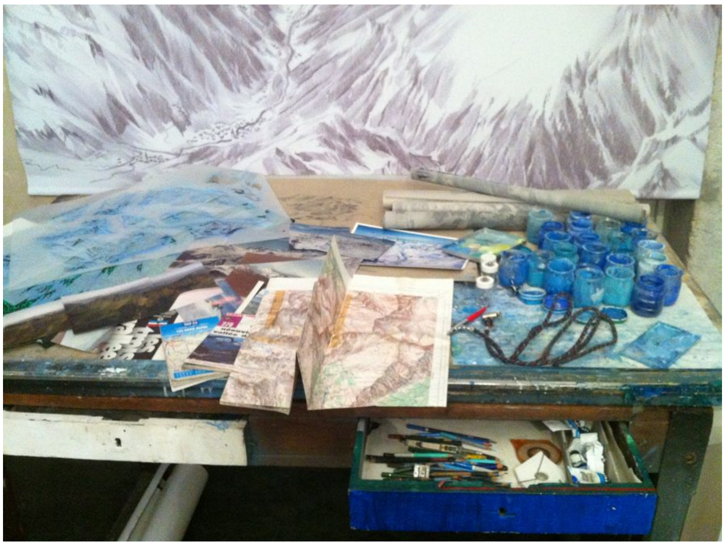


Figure 2.19: Laying out photographs and maps for proper deformation of terrain.

The painting process for a single panorama map is fairly elaborate, time-consuming and roughly takes six-eight months for completion from the time of inception. Following are the steps that are taken to create a panorama painting :

- The starting point is to take a map from the TOP25 "Institut national de l'information géo-graphique et forestière" (IGN, french governing body for cartographic mapping and surveying of mountainous regions and forest cover). Arthur Novat then proceeds to understand the terrain layout and chooses a suitable viewpoint for the panorama.

- This is followed by a brief survey of the surrounding landscape with photographs (survey plane, satellite imagery etc.) The point of this exercise is to understand the terrain's physiognomy.
- The sponsor's request is then handled and Novat decides an appropriate method to *unfold* (deform) the landscape to best suit their needs (refer Figure 2.19).
- An initial pencil sketch is created (refer Figure 2.20) that contains shaded elements without hue information or surface texture.
- The pencil sketch is then validated with the sponsor's requirements.
- In case the sketch is approved, Novat proceeds to color the given sketch :
    - The coloring tools used are mainly crayons, color pencils, gouache and *gesso* acrylic.
    - The pencil layer is off-set with a color pencil layer.
    - Supports are added to edges and protruding areas with color pencils.
    - Masking is done by applying a layer of airbrush with gouache using rhodoids.
- Final terrain details (surface textures) are filled in :
    - Fir trees : color pencil
    - Rocks and boulders : Paint and color pencils
    - Houses and settlements : Paint and color pencils
    - Whitening (snow, highlights etc.) : Gouache layer (wash) or reveal white, base acrylic layer through scratching.

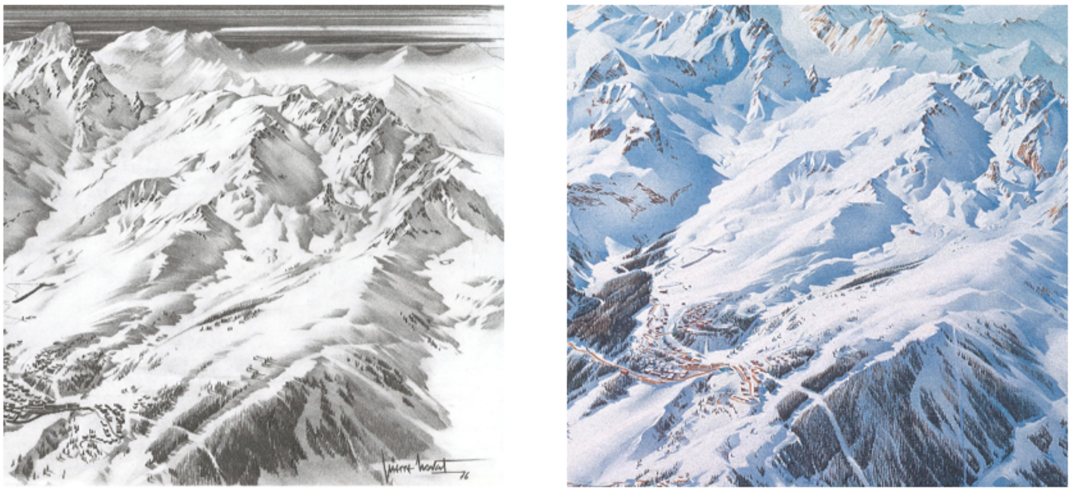


Figure 2.20: Pencil sketch of terrain (left), colored variant of same terrain (right), Courchevel, 1976

2.3.3 Artist-specific visual cues

The panoramas created in the Novat-style contain several unique visual cues that are useful for interpretation. Some are decorative while others play are significantly more important. The main visual elements can be described as follows :

- **Shadows and Illumination** : Variations in lighting allow the viewer to understand the shape of the mountain. In some cases, upon close observation, there appear to be lighting from different directions (non-realistic and highly improbable as only the sun is a primary source). However, this is intentionally done by the artist to highlight certain mountain shapes.

  Shadows are slightly more complicated; cast shadows, shaded areas on the leeward side of relief and shadows for individual terrain elements, each play a role in viewer perception. The shadows act as visibility tests and more often than not, allow the viewer to discern scene depth. Additionally, shadows also have a difference in gradient with light, white spots/bands, indicating inter-reflections (light bouncing off of surrounding mountain faces).

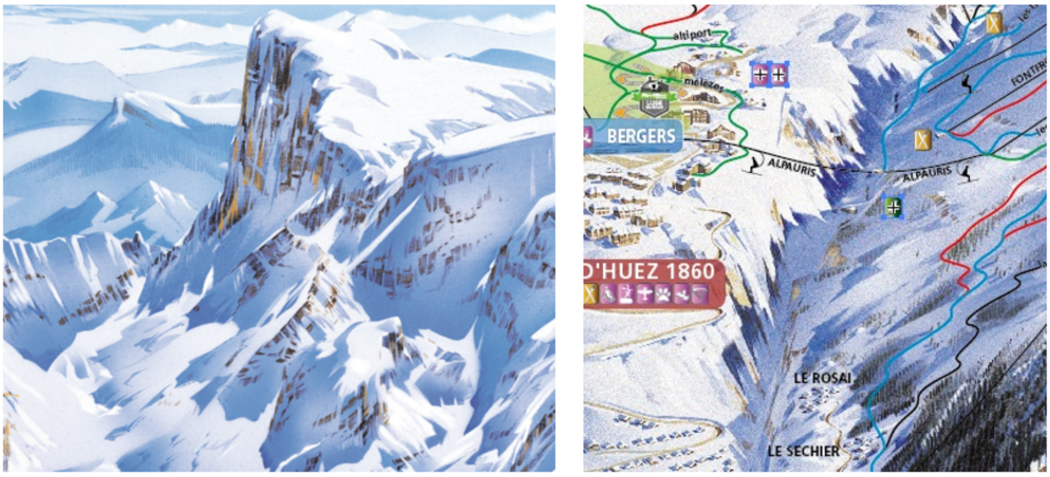


Figure 2.21: Cast shadows with non-uniform color gradient (left), shadows extending past valley boundaries(right)

- **Snow** : Changes in hue depending on type of snowpack present on mountains (glacial snow, fresh powder snow and ice), both in terms of altitude and aerial perspective.
- **Forest cover** : Provide cues in the form of organization and alignment; the direction of the clump along with the color determines the shape and region of the mountain face. Colors usually range from light green to dark green with occasional white fir trees.
- **Rocks and boulders** : Less important than the previous elements but serve as an indicator for dangerous areas (steep, rough slopes with dark coloration). Can also be used as a decorative element or as a potential landmark. Colors usually range from light-grey to light-brown and the shape varies in accordance with the mountain shape.

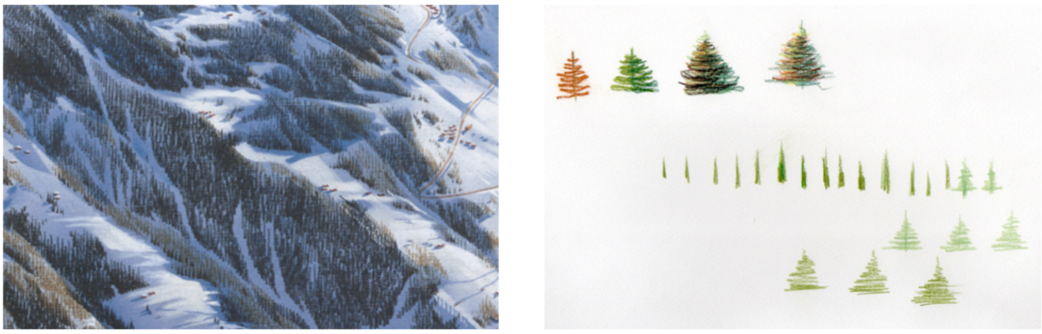


Figure 2.22: Clump of trees facing downward along the slope (left), individual tree textures (right)

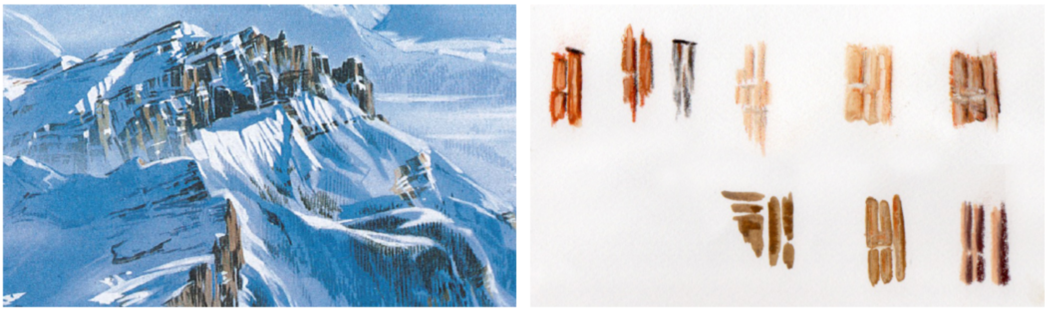


Figure 2.23: Rock formation protruding from a mountain slope (left), individual rock textures (right)

- **Houses and settlements** : Purely decorative and serve as a point of reference for valleys and popular resort-spots. Shapes are usually minimalist and colored brown with white roofs.

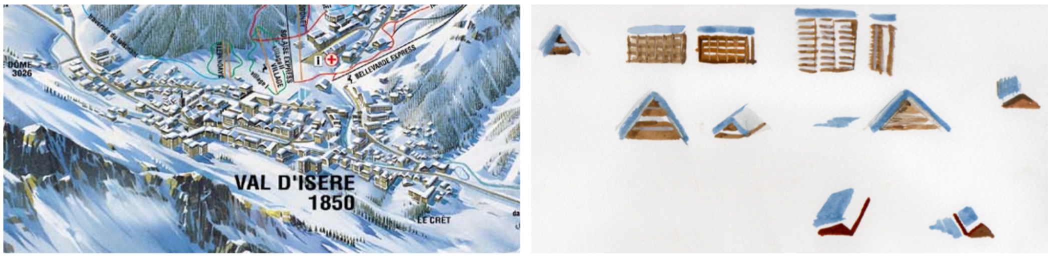


Figure 2.24: A village cluster in Val D'Isere (left), individual house textures (right)

- **Road-lines** : Drawn in a continuous line and follow along existing roadways. Usually represented as engraved hollows in the snow.

- **Mountain-scape backdrop** : Consists of outlines of mountains that are drawn in a highly simplified manner with much less detail. Showcases the effect of aerial perspective with increasing distance from the viewpoint.

- **Sky** : Occupies the top 30-40 % of the panorama. Light blue accents at the bottom, near the mountain peaks and a gradient that becomes darker with height.

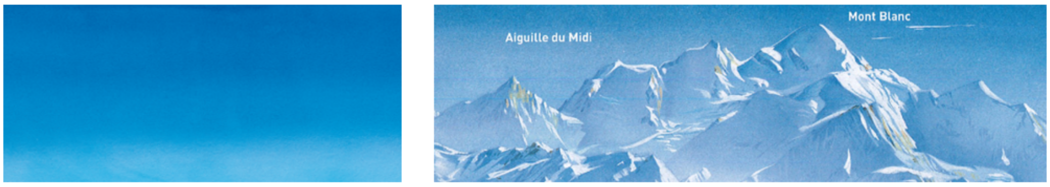


Figure 2.25: Sky gradient (showcasing shift in hue) (left), mountain-scape backdrop (right)

2.4 Related work

In terms of related work, there has not been much done in the context of panorama maps. In fact, there are only a handful of papers that focus on rendering panorama maps over the past decade. This section will focus on two specific papers put forth by [BS17] and [BST09]. Both of these papers focus on methods for rendering panorama maps painted by the late Austrian painter and cartographer, Heinrich Caesar Berann. In the context of this thesis and from the literature surveyed, there is no significant work done on replicating/rendering panorama maps made by the Novat style. Nevertheless, some inspiration can be drawn from these two papers.

[BST09] propose a pipeline to accurately generate (up to an extent) panorama maps in the style of H.C. Berann. In order to do so, they derive a set of heuristics and principles that would allow them to express the stylistic preferences of these artists. Moreover, panorama design principles from the American cartographer and painter, James Niehues are taken into consideration as well. In terms of visual elements, five categories of terrain textures are considered; trees (evergreen and deciduous), cliffs, snow, lakes and grass. These are used in the form of 3-D textures that are placed depending on the surface contours of the terrain. A custom shading model is also used with a predefined, directional light source. These are preceded by a geometry deformation step wherein the base plane of the terrain model is bent to showcase *progressive perspective* [Pat00]. Additionally, a vertical exaggeration step is added to enhance high elevation areas while keeping rest of the terrain intact.

A more recent method proposed by [BS17] takes a step forward from the method by [BST09], wherein, the analysis of Berann's panoramas are used in a real-time context with real-world terrain data. A pipeline is thus proposed that takes into consideration the terrain elements (trees, cliffs etc.), color palette (local and global), changes in intensity, planar distortion and algorithms for rendering clumps of trees along with accurate water reflections.

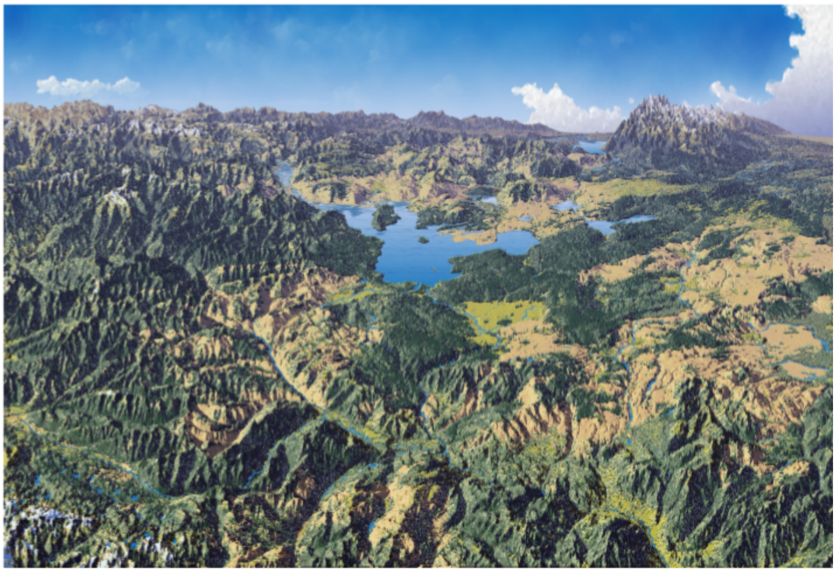


Figure 2.26: A panorama map of the Yellowstone National Park produced by the algorithm proposed by [BST09]

An interesting point to note is that both the methods described above do not perform any drastic modifications with respect to illumination or shadows, but use minor layering on-top of the standard shading model. The assumption being : both methods focus on replicating summer-time panoramas by H.C. Berann. In these panoramas, there is a heavy emphasis on surface textural elements and extensive stylization with respect to the shading model; a stark contrast when compared to it's winter counterpart. Therefore, the methods proposed in this thesis aim at the core-aspect of winter-panorama maps : cast shadows. Additionally, with no prior work done on panorama map rendering in the style of Atelier Novat or for winter-panorama maps, the rendering framework proposed in this thesis aims to bridge this knowledge gap.

3 | Methodology

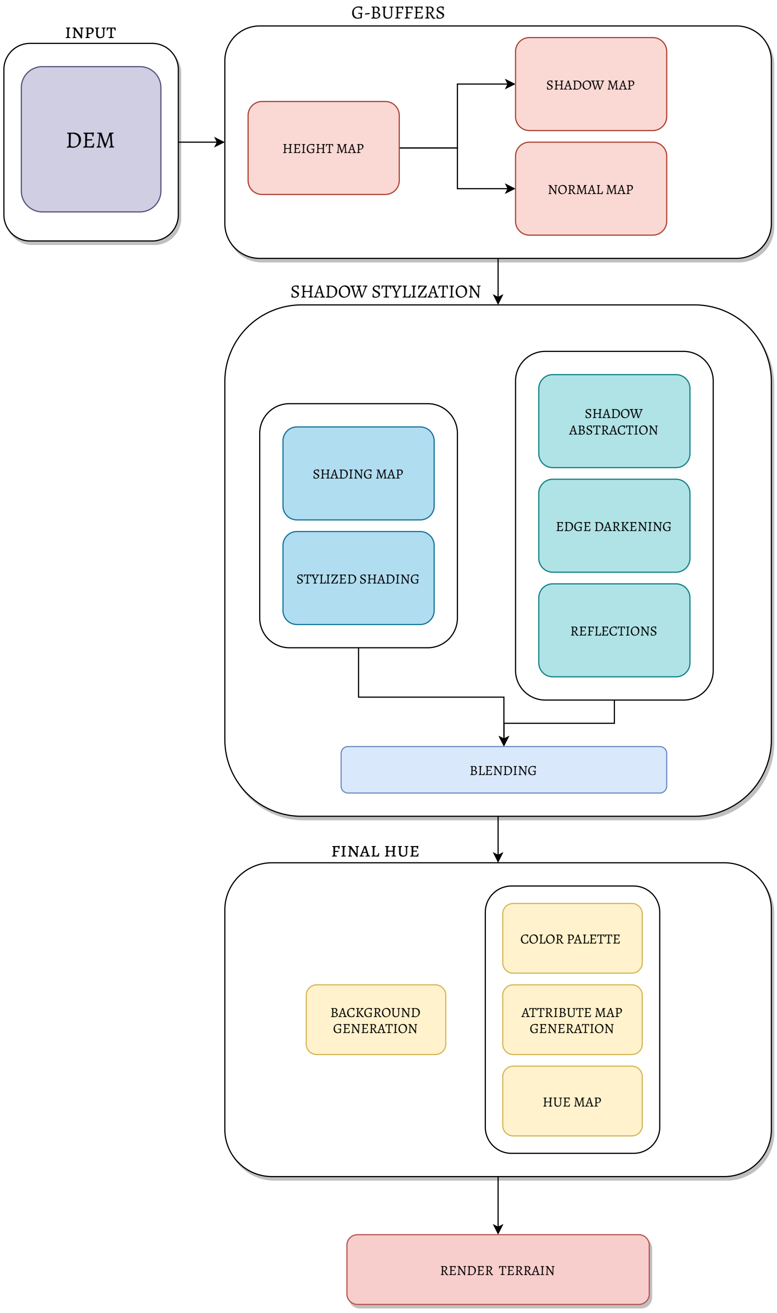


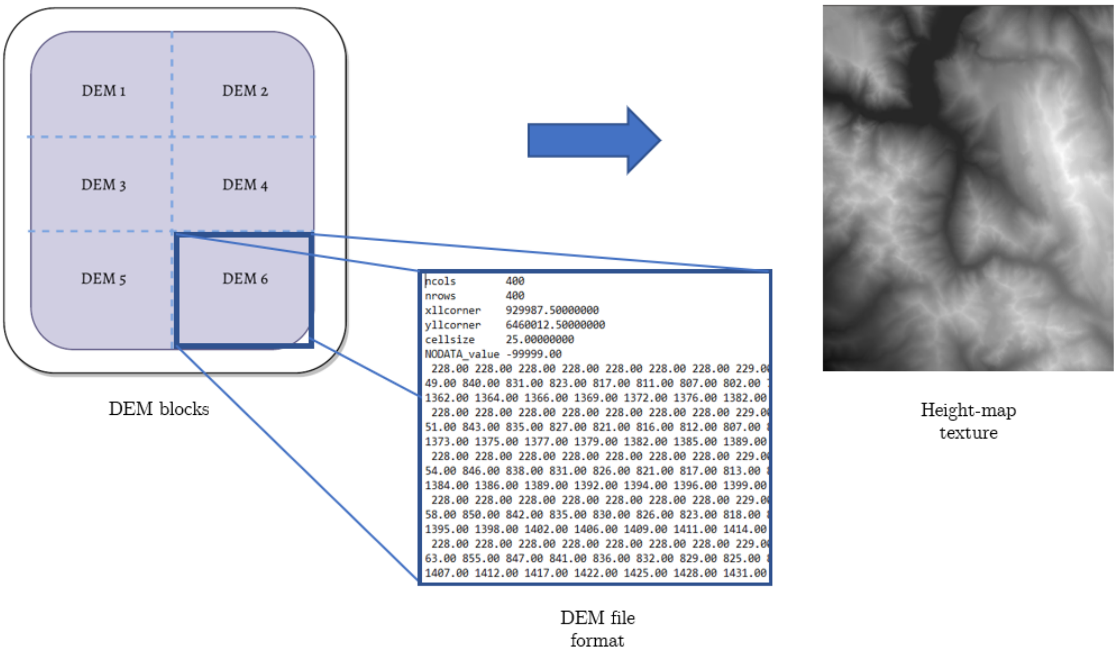
Figure 3.1: A holistic overview of the proposed rendering framework

This chapter will mainly elaborate on the methods used in the proposed rendering framework (see Figure 3.1) for stylizing cast shadows in panorama maps, drawn in the style of Novat. The chapter outline will follow the flow of diagram 3.1; DEM to height-map texture conversion is tackled first followed by details on the algorithms used to stylize both the shading map and shadows. Finally the section ends with a brief overview of software specifications that was used to implement the framework. To note, the algorithms discussed later on in this section are pseudocode representations of actual GLSL code that is implemented in fragment shaders.

3.1 Rendering pipeline

To recapitulate from earlier chapters, the methods used in the pipeline operate entirely in image space. In fact, the cast shadows and shading model are stylized through image-filtering methods. Every single sub-step within the pipeline uses preexisting information (normals, heights, visibility) in the form of textures through g-buffers. Additionally, all operations except for DEM-to-heightmap conversion (CPU) take place on the GPU, accessible through fragment shaders.

3.1.1 DEM-heightmap conversion



```
ncols        400
nrows        400
xllcorner    929987.50000000
yllcorner    6460012.50000000
cellsize     25.00000000
NODATA_value -99999.00
 228.00 228.00 228.00 228.00 228.00 228.00 228.00 229.0
49.00 840.00 831.00 823.00 817.00 811.00 807.00 802.00
1362.00 1364.00 1366.00 1369.00 1372.00 1376.00 1382.00
 228.00 228.00 228.00 228.00 228.00 228.00 228.00 229.0
51.00 843.00 835.00 827.00 821.00 816.00 812.00 807.00
1373.00 1375.00 1377.00 1379.00 1382.00 1385.00 1389.00
 228.00 228.00 228.00 228.00 228.00 228.00 228.00 229.0
54.00 846.00 838.00 831.00 826.00 821.00 817.00 813.00
1384.00 1386.00 1389.00 1392.00 1394.00 1396.00 1399.00
 228.00 228.00 228.00 228.00 228.00 228.00 228.00 229.0
58.00 850.00 842.00 835.00 830.00 826.00 823.00 818.00
1395.00 1398.00 1402.00 1406.00 1409.00 1411.00 1414.00
 228.00 228.00 228.00 228.00 228.00 228.00 228.00 229.0
63.00 855.00 847.00 841.00 836.00 832.00 829.00 825.00
1407.00 1412.00 1417.00 1422.00 1425.00 1428.00 1431.00
```

Figure 3.2: Conversion from structured DEM blocks to a height-map texture

In order to construct/render a terrain scene, a set of height values is required. In this case, the data consists of six DEM blocks, structured according to intercardinal directions (north-east, north-west, east, west, south-east, south-west; from right to left, top to bottom). Typically, a DEM file is a simple, regularly spaced grid of elevation points and the most common format for terrain distribution.

From Figure 3.2, the DEM file contains two parts : a header and set of elevation points (in real-world heights). In order to represent it as a greyscale texture, the file is parsed by separating the header and elevation points. From the header, the dimensions along with the cell-size is obtained. The elevation points are then normalized between the range [0..1] and then *stitched* together with other DEM blocks to form a collective matrix of size 1200 x 800 (height x width) with normalized height values. The matrix is then visualized and stored as a height-map texture, to be used further on in the pipeline.

3.1.2 G-buffers

The G-buffer is a collective term for all textures used to store lighting-relevant data for the final lighting pass in deferred rendering [1]. In this specific case, the G-buffer comprises of a normal map, shadow map and the height map textures. Contrary to the standard G-buffer definition that captures the information from the camera, we exploit the fact that the terrain is 2-D and employ the G-buffer as additional layers to the height field. These texture are calculated and stored as follows:

- **Normal map** : Basically a 2-D texture that stores surface normals. It's main usage lies in lighting calculations and enhancing surface details without adding extra polygons to meshes. In this case, calculating normals is fairly straightforward : the normalized cross product between two positional vectors on the height map gives the surface normal at that point. (Figure 3.3 showcases the normal map being used.)

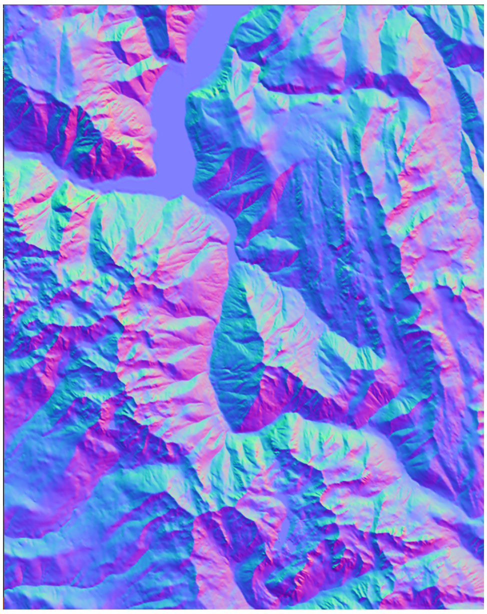


Figure 3.3: Normal-map texture from corresponding height-map texture with strengthened surface normals

[1] https://learnopengl.com/Advanced-Lighting/Deferred-Shading

- **Shadow-map**: Shadow-maps are essentially visibility indicators that state whether a point is occluded in the path of light. There are a lot of different techniques for generating shadow maps [WPF90]. The focus however, will be on hard-shadows, specifically obtained by ray-marching. Ray-marching is an interesting technique wherein a path is "marched" from a point, in the direction of the light source (see Figure 3.5). The advantage of using such a technique with height-maps is twofold; first, it is a really simple technique in principle. Second, going step-by-step in the direction of the light source is intuitive considering height information available at each point.

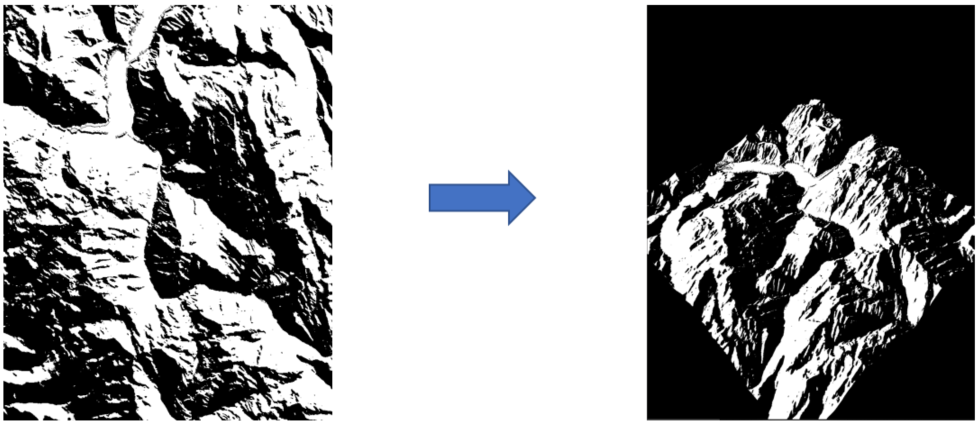


Figure 3.4: ray-marched shadow-map texture with a light source (left), texture mapped onto terrain (right)

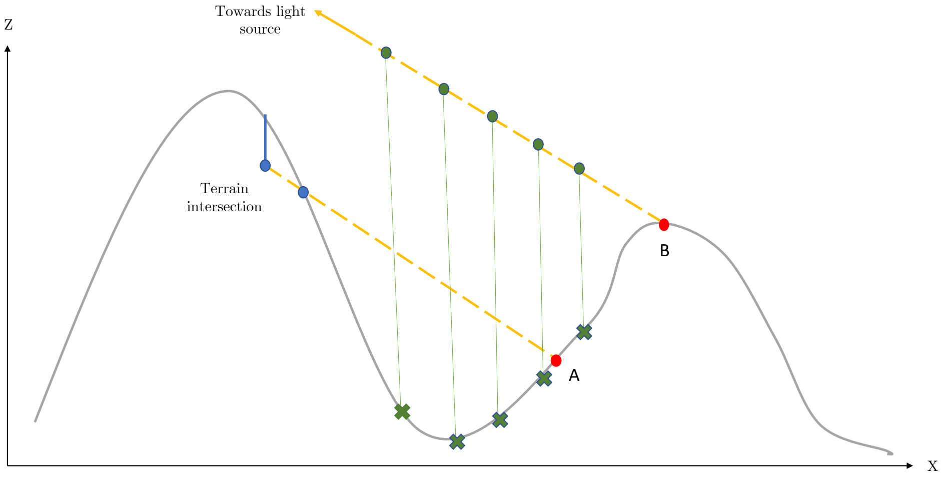


Figure 3.5: Diagram showcasing raymarching with two example points on height-map (A, B)

Algorithm 1: Ray-marched shadows

```
Input: height-map
Output: ray-marched shadow map
$currentHeight := startHeight$
$stepCoord.xyz := (textureCoord, currentHeight)$
while $i < textureWidth$ do
    $stepCoord.xy += lightPos.xz * pixelLength$
    $stepCoord.z += lightPos.y * \sqrt{1 + (\frac{1}{pixelLength.x})^2 + (\frac{1}{pixelLength.y})^2}$
    if $currentHeight < heightMap$ then
        $result := 0$
        $exit$
    end
    $currentHeight := stepCoord.z$
end
$result := 1$
```

Notation: $currentHeight$ represents the initial height of the pixel being evaluated with $startHeight$ as the actual value of the point on the height-map texture. $stepCoord$ is the 3-D increment vector having the current texture-coordinate (XY in 2-D) and the height along Z-axis. $lightPos$ is the 3-D vector indicating the position of the light source and $pixelLength$ is the pixel-wise increment along XY-axis in 2-D texture-space. $heightMap$ represents the height-map texture that can be indexed by the texture-coordinate $texCoord$ or by the increment vector $stepCoord$.

Shadows through raymarching can be explained through Figure 3.5 and the algorithm above. Rays that are marched step-by-step in the direction of the light source from the points on the height-map (A and B), either intersect with the terrain or travel indefinitely until an exit condition is met. In case of terrain intersection, a value of 0 is returned as the pixel-color of the point through which the ray originates. Otherwise, a value of 1 is returned for the latter condition.

3.1.3 Stylized shading

For the shading model, a basic Lambertian shading (diffuse) is used. This is mainly due to the fact that ambient and specular components (with respect to lighting) do not play a role when the material being considered is snow-cover on top of a mountain-scape (The approximation of a rough-diffuse surface works well in this case [War82]). Additionally, the artist does not appear to paint glints and highlights (specular) on the surface, rather, it appears to be smooth and consistent.

In terms of stylization, a novel shading method is proposed that aims to mimic the shading done by Novat in the ski-panorama maps. To reiterate on the painting process, the Novat style dictates the use of watercolor-pencils to emphasize mountain edges, lines and crevices. Additionally, large mountain faces are usually smoothed by brushing the pigment inwards from the edges, thereby, creating a diffused, flow-like effect. This is usually layered with gouache or acrylic to produce highlights and to raise pencil-colored mountain-lines.

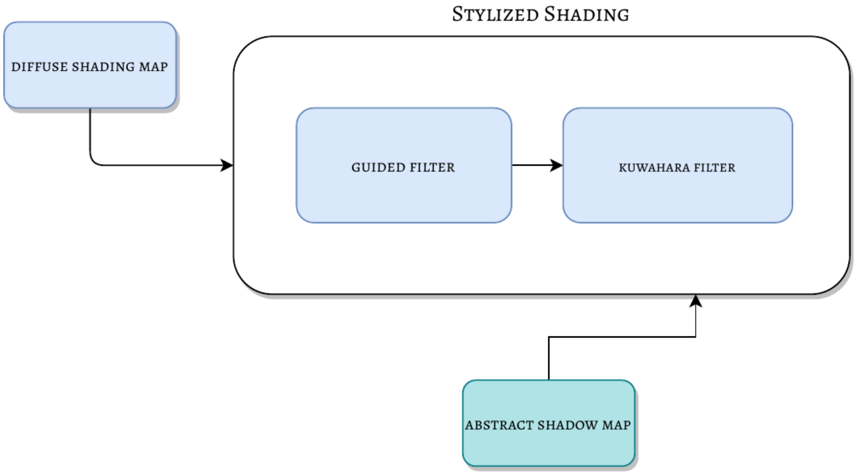


Figure 3.6: Stylized shading sub-steps

To mimic the approach described above, the proposed method consists of two steps applied one after the other in succession (as shown in Figure 3.6) and consists of two edge-preserving smoothing filters; the guided filter [HST10] and a slightly modified version of the Kuwahara filter [Kuw+76]. Although these will be discussed in detail in the next chapter, a brief introduction can be given as follows.

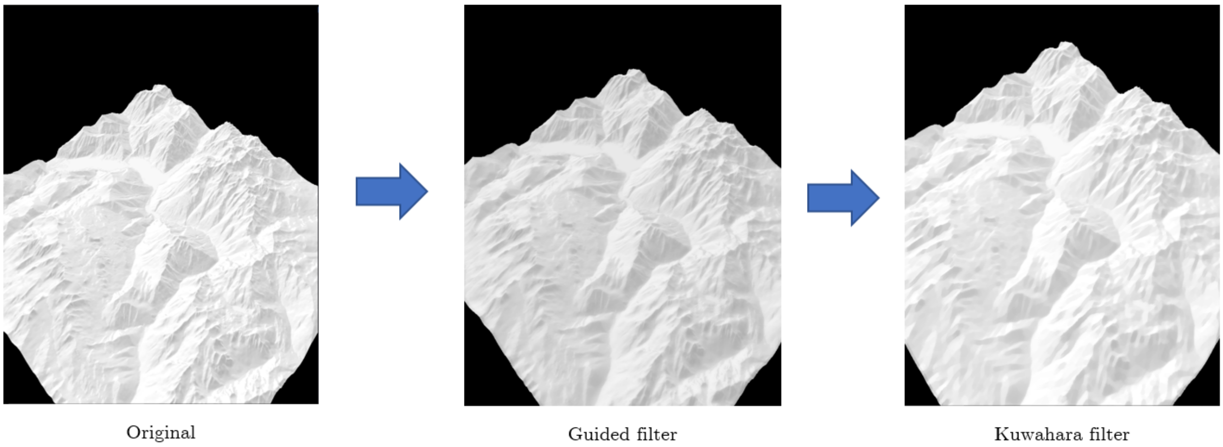


Figure 3.7: Iterative application of the guided filter and kuwahara filter on the original diffuse shading map.

The guided filter aims to mimic the first half of the artistic shading process, wherein, edges are emphasized while rest of the elements are smoothed. The kuwahara filter on the other hand (well known for simulating oil and acrylic like effects [GDG19]) aims to apply brush-like strokes to these emphasized edges while retaining smoothed effect on the other regions. Additionally, a condition is added whereby the size of the kuwahara filter kernel changes depending on whether the region is in shadow or under illumination. The algorithms for both guided and Kuwahara filter can be given as follows:

Algorithm 2: Guided filter implementation

```
Input: Input image $p$, guiding image $I$
Output: filtered output image $q$
$mean_I := f_{mean}(I, r)$
$mean_p := f_{mean}(p, r)$
$corr_I := f_{mean}(I. * I, r)$
$corr_{Ip} := fmean(I. * p, r)$
$var_I := corr_I - mean_I. * mean_p$
$cov_{Ip} := corr_{Ip} - mean_I. * mean_p$
$a := \frac{cov_{Ip}.}{(var_I + \epsilon)}$
$b := mean_p - a. * mean_I$
$mean_a := f_{mean}(a, r)$
$mean_b := f_{mean}(b, r)$
$q := mean_a. * I + mean_b$
```

Notation: As noticeable from the guided filter algorithm, the guided filter is essentially a computation of a series of box filters. I, p, q are the guiding, input and output image respectively. $f_{mean}(., r)$ indicates a mean filter with radius r; cov, var and $corr$ indicate co-variance, variance and correlation respectively.

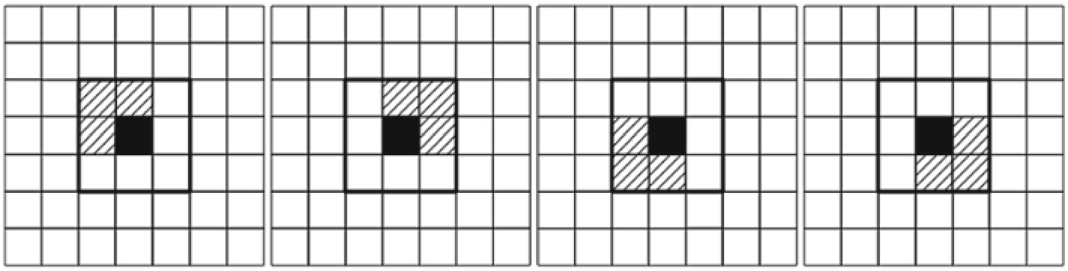


Figure 3.8: Kuwahara filter rotation for $3x3$ kernel as described in [Bar16]

Notation: The Kuwahara algorithm is described for a center pixel in a kernel of size n x n, wherein the original algorithm is tweaked to hold a flexible kernel size ($sizeLit$, $sizeShadow$) depending on the location of center pixel. This can be observed in Figure 3.9 wherein, the area bounded by the shadow shows is fairly smooth (with the absence of stroke lines), whereas, the region that is lit has exaggerated stroke lines indicating the mountain slope. In some paintings, this effect is reversed or both areas are kept smooth. Therefore, in order to address these effects, a flexible, region-dependent kernel is needed.

Algorithm 3: Kuwahara filter implementation

```
Input: Input image, kernel size ($n$), shadow-map texture
Output: filtered output image $q$
if $shadowMapColor == 1$ then
    $n := sizeLit$
end
else
    $n := sizeShadow$
end
$k := n + 1$
while $i < k$ do
    $mean_k := \frac{1}{(n+1)*(n+1)} \cdot \Sigma_{(x,y)\epsilon\theta_k} \phi(f(x,y))$
    $var_k := \frac{1}{(n+1)*(n+1)} \cdot \Sigma_{(x,y)\epsilon\theta_k} [\phi(f(x,y)) - mean_k]^2$
end
$result := mean_i | var_{min} \epsilon Min(var_k)$
```

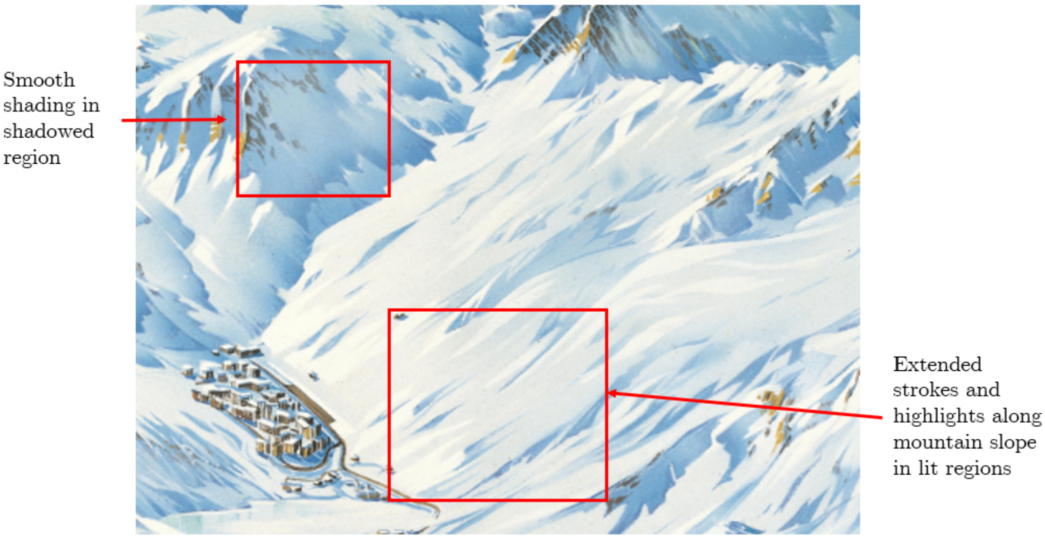

Smooth shading in shadowed region
Extended strokes and highlights along mountain slope in lit regions

Figure 3.9: Cropped from Espace Killy, Pierre Novat, 1983. Showcasing difference in shading across shadowed and lit regions (Emphasis is shown by the red boxes).

Continuing the notation, k is the number of sections within the filter, also shown in the Figure 3.8. The end result is basically the mean value of the section containing the smallest variance. θ_k represents each section individually and $\phi(f(x, y))$ is the value at that particular pixel.

3.1.4 Shadow stylization

This section mainly deals with three methods for stylizing cast shadows in the style of Novat, these are : shadow abstraction, edge-darkening of shadows and inter-reflections. The first two effects draw inspiration from the methods described in Chapter 2.

- **Shadow abstraction** : Abstraction is an important parameter to consider when studying the style of Atelier Novat. This holds true especially in the case of cast shadows. In the panorama paintings, shadows are seen as rounded and having smooth boundaries (while being sharp and well-defined). This allows the viewer to extrapolate a number of visual cues: the presence of rounded, highly abstract shadows in a particular region may indicate a ski-safe region. Additionally, increasing the level of abstraction, also increases the non-realistic aspect of these paintings.

  In regards to the method used, there are a number of ways to achieve varying level of abstraction. This thesis proposes a two-step approach (inspired by the work of [LD05]) wherein, the shadow-map texture is first convolved with a Gaussian kernel (increasing kernel size leads to higher abstraction), after which, a threshold is applied to binarize the blurred shadow-map texture. This has an effect similar to binary morphological dilation [LD05] (see figure 3.10), thereby resulting in large, abstract, binary regions.

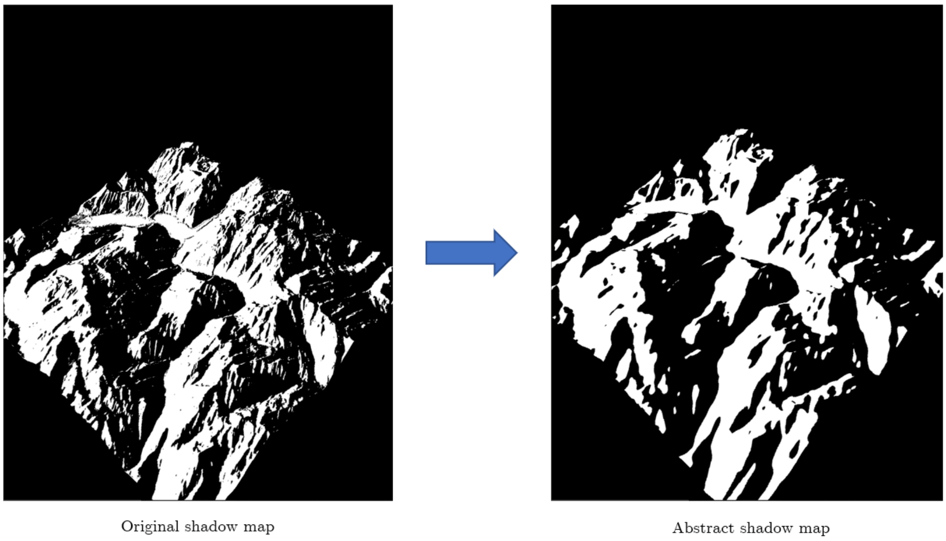


Figure 3.10: Abstract shadow map generation

- **Edge darkening** : Unlike traditional applications of watercolor, where this effect is far more prevalent, the Novat style uses edge darkening as means of contrast-enhancement. To put it in a simpler manner, this effect is akin to *unsharp-masking*, which is a digital-image processing technique that uses a blurred, inverted image as a mask for increasing local contrast in images (an example is shown in Figure 3.11 [2]).

[2] https://en.wikipedia.org/wiki/Unsharp_masking

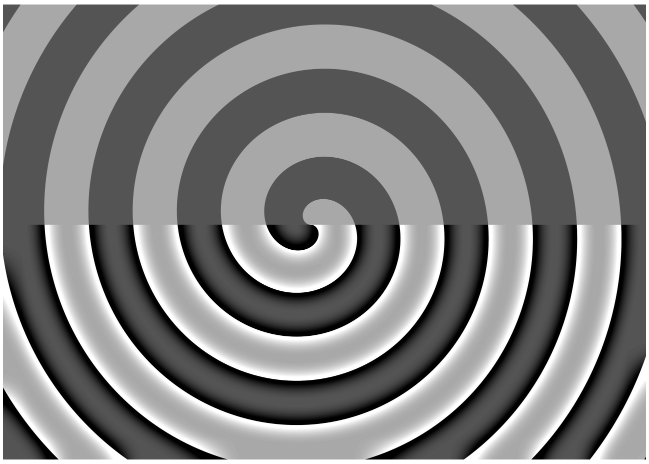


Figure 3.11: An example of unsharp masking

The edge-darkening effect mentioned here takes some inspiration from this technique, however, the difference is usually in the thickness of the bands appearing at the edges as well as the relative contrast between the center of the shadow and the edge(see Figure 3.12).

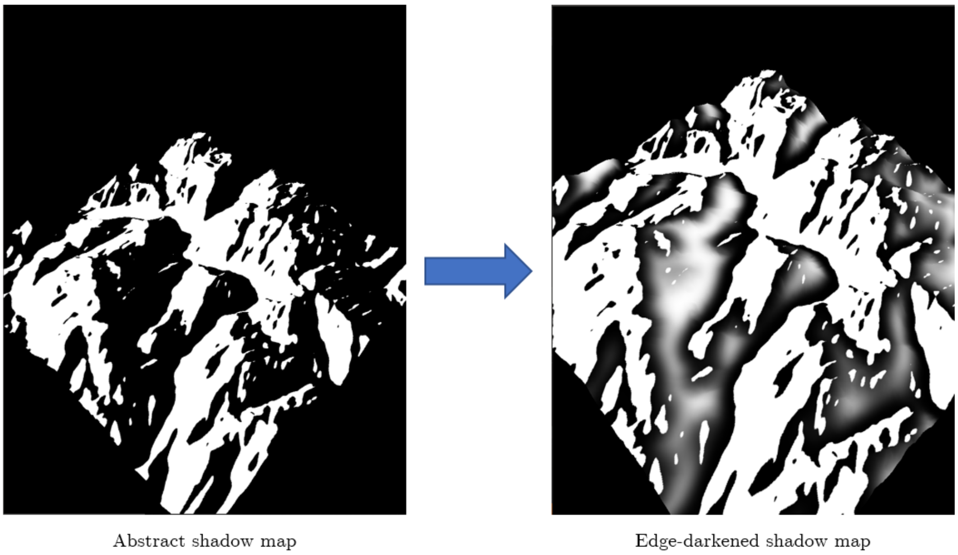


Figure 3.12: Edge-darkened shadow map generation

Additionally, the edges that are darkened in the panorama maps vary depending on the height of the mountain slope that the shadow rests on. For example, given a specific shadow region, the top band may appear thicker as compared to the bottom, resulting in a shift of perception. The algorithm proposed offers the user a set of parameters ($clamp_{upper}$ and $clamp_{lower}$) that allow shifting of these bands (darkened-edges) albeit on a global level (explained further in Chapter 4).

```
Algorithm 4: Edge darkening in shadows
    Input: abstract shadow-map, height-map, blurred shadow-map
    Output: edge-darkened shadow-map
    $counterShade := abstractShadowMap + (1 - blurShadow)$
    $counterShade := clamp(counterShade, 0, 1)$
    $\gamma := clamp_{upper} / (height + clamp_{lower})$
    $result := clamp(counterShade^{\gamma}, 0, 1)$
```

- **Reflections** : Inter-reflections form one of the most important cast-shadow effects, mainly due to the information that they convey on a mountain face; inter-reflections are used as visual cues to understand the positioning of surrounding mountain faces. Stronger, wider reflections along a region in shadow may indicate the presence of a long mountain range opposite to the shadowed face. Additionally, along with edge-darkening, inter-reflections also contribute to the counter-shading effect.

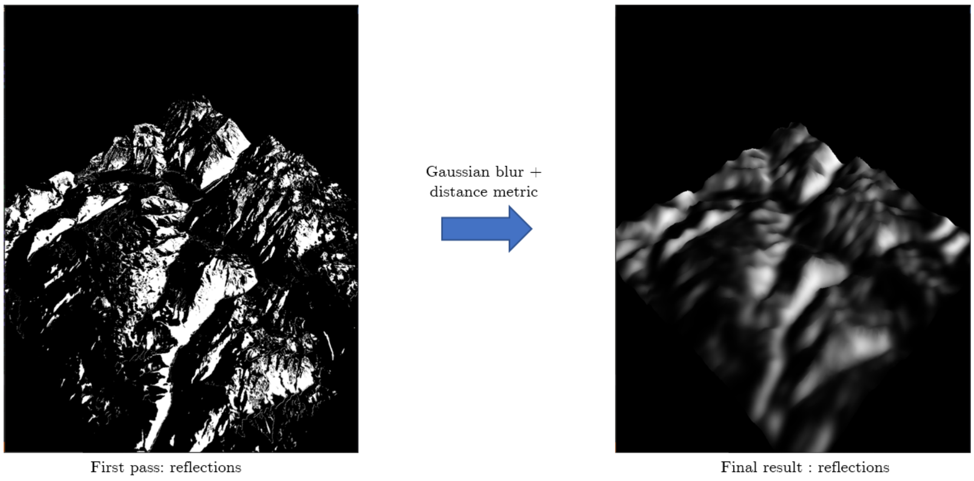


Figure 3.13: Inter-reflections shadow map generation

The proposed method for calculating inter-reflections borrows a few ideas from physically-based rendering. In fact, the approach is similar to a simplified, pseudo-version of global illumination. The surface normals facing a certain direction in shadowed regions are projected along the horizontal plane and ray-marching is done to check the nearest intersection point. Upon finding the point, the value stored in the shadow-map for the intersection point is queried and thereby set as the value of the pixel at the ray-origin (refer Figure 3.14). The algorithm is as follows :

Notation: The symbols above are similar to the ray-marching algorithm 1, wherein $abstractShadowMap$, $heightMap$ are the abstracted shadow map and height map respectively. The other symbols follow the same notation as mentioned in algorithm 1.

Algorithm 5: Inter-reflections in shadows

```
Input: abstract shadow-map, height-map, normal map
Output: inter-reflection map
if $abstractshadowmap == 0$ then
    $currentHeight := startHeight$
    $newNormal.xyz := (normal.x, 0, normal.z)$
    $stepCoord.xyz := (textureCoord, currentHeight)$
    while $i < textureWidth$ do
        $stepCoord.xy += normal.xz * pixelLength$
        if $currentHeight < heightMap$ then
            $result := clamp(abstractShadowMap.x/distance, 0, 1)$
            $exit$
        end
    end
    $result := 0$
end
else
    $result := 0$
end
```

Additionally, after generating the inter-reflection map, it needs to be processed further; a Gaussian blur with a large radius is applied in order to smooth the noise. A distance-based parameter can also be set (given by *distance*) that allows points to be filtered based on the distance traveled by the ray; the shadow-map value queried at the intersection point is divided by the normalized distance between the points. The assumption is that, greater the distance traveled by the ray, lesser is the contribution towards reflection.

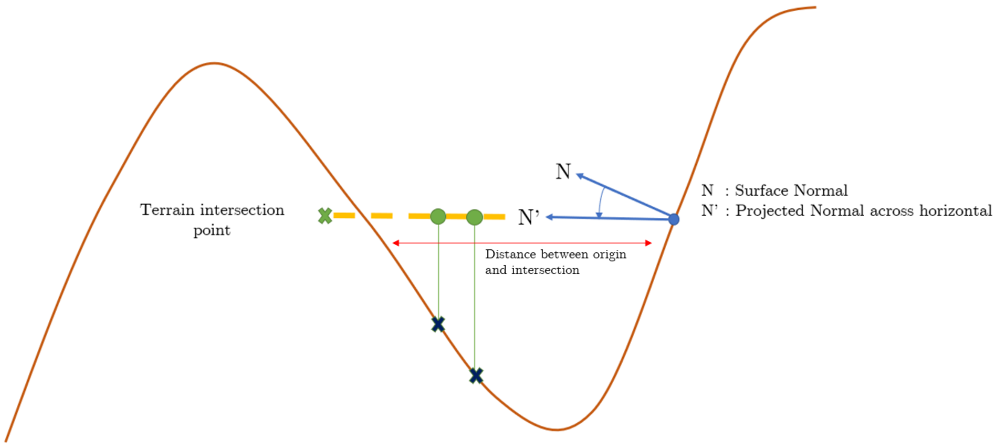

Terrain intersection point
N
N'
N : Surface Normal
N' : Projected Normal across horizontal
Distance between origin and intersection

Figure 3.14: Diagram showcasing projected normals marching along the horizontal

3.1.5 Blending

After computing and storing the stylized shading map and the stylized shadow maps, the question of combining both of them (to represent a single intensity map) remains. The approximation considered here is the fact that shading and shadows can be decoupled. This can be shown through the rendering equation (see equation 3.1) given by [Kaj86] :

$$L_o(p,\omega) = L_e(p,\omega) + \int_{\Omega_+} f_r(p,\omega,\hat{\omega})L_i(p,\hat{\omega})cos(n_p,\hat{\omega})d\hat{\omega} \tag{3.1}$$

Where p is the scene point, n_p is the surface normal, $(\omega,\hat{\omega})$ are the reflected and incoming light direction respectively, Ω_+ denotes the hemisphere around the point p. L_o, L_e, L_i represent the outgoing, emitted and incident radiance respectively, with f_r being the BRDF (Bidirectional Reflectance Distribution Function) that describes the portion of light that is reflected in a certain direction at point p.

With that said, the Equation 3.1 can be further simplified by making certain assumptions thereby obtaining the *direct-lighting equation*. Direct-lighting can be defined as the amount of light falling on an object from a given light source (shown in Figure 3.15[3]). Additionally, depending on the surface orientation towards the light source, the amount of light reflected may change. Therefore, the approximate local behavior of light on the object's surface relative to it's orientation can be termed as "shading" (effectively quantified through the BRDF in Equation 3.1). With that said, the following assumptions can be made about illumination and light source:

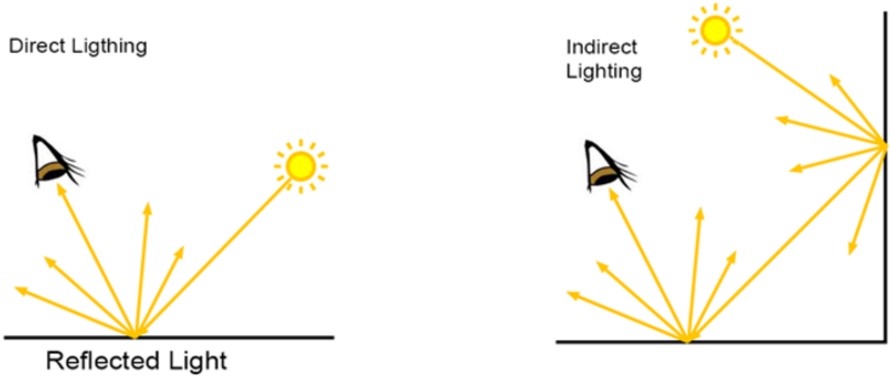


Figure 3.15: Diagram showcasing direct (left) and indirect (right) lighting

[3] https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading

- L_e can be set to zero, as our scenes do not contain emitting surfaces.
- Since we are interested purely in direct illumination, for all points q, the outgoing radiance, L_o is zero except for points lying on the light source.
- There is only one light source in the scene.

$$L_o(p,\omega) = \int_L f_r(o,\omega,p \to l)L_e(l,l \to p)G(p,l)V(p,l)dl. \tag{3.2}$$

In practice, the Equation 3.2 can be further simplified to showcase shadow-shading decoupling :

$$L_o(p,\omega) = \int_L f_r(p,\omega,p \to l)G(p,l)dl \cdot \frac{1}{|L|}\int_L L_e(l,l \to p)V(p,l)dl \tag{3.3}$$

As seen from Equation 3.3, the first integral represents represents shading, while the second represents a shadow. There are two assumptions made: first, if the distance from light source to scene is large and the shape of light is simple, then the geometric term G shows little variance. Secondly, the BRDF f_r is considered to be mainly diffuse. Therefore, splitting the integral into the product of two integrals around G and L_e is a good approximation.

Algorithm 6: Blending condition for final intensity-map

```
Input: abstract shadow-map, edge-darkened shadow-map, reflection
       shadow-map, stylized shading-map
Output: Intensity map
if $abstractShadowMap == 1$ then
    $Color := shadingMap^{tone_{lit}}$
end
else
    $Color := (shadingMap^{tone_{shadow}}) \cdot clamp(\alpha_{edegeDark} \cdot$
      $(edgeDarkMap^{tone_{edgeDark}} + reflectionMap), 0, 1)$
end
```

From Algorithm 6, the final color of a pixel present on the intensity map can be calculated depending on whether it lies in shadow or lit region. A non-linear operation is performed to control luminance of shadow, shading maps by considering three user-controllable parameters : $tone_{lit}$, $tone_{shadow}$ and $tone_{edgeDark}$. The parameter $\alpha_{edegeDark}$ can also be set by the user to control overall brightness of the edge-darkening map.

The main goal in using the parameters mentioned above is to control the white tone-level within the shadowed and lit regions. From Figure 3.16, it can be seen within the combined stylized-shadow map that there are varying tones of white. The objective here, is to increase white-tone level in the inter-reflection map while decreasing it in the edge-darkened map. This ensures priority given to reflections in the final intensity map. Simultaneously, the edges are supposed to be darkened as well. However, this can be changed depending on the type of style to be mimicked. The explanation here works for a general case observed in most panorama maps.

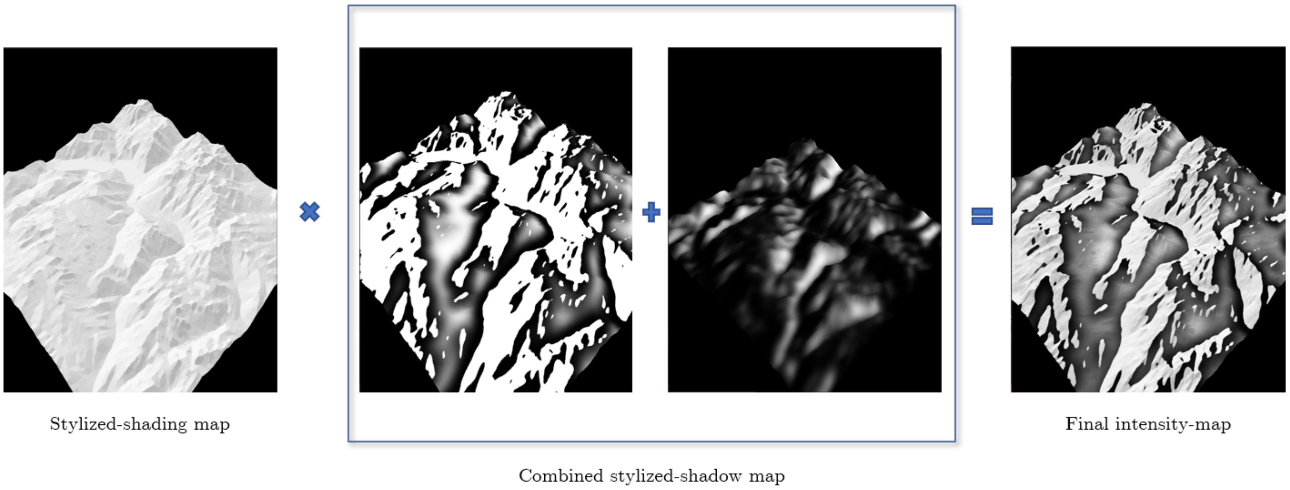


Figure 3.16: Diagram showcasing blending condition for final intensity-map

3.1.6 Hue calculation

After calculating the intensity-map, filling it with color is the next logical step in the pipeline. With regards to the hue, there are several *zones* in the panorama maps that showcase a different variation to the hue norm. These are : foreground hue, background hue and sky gradient (shown in Figure 3.17).

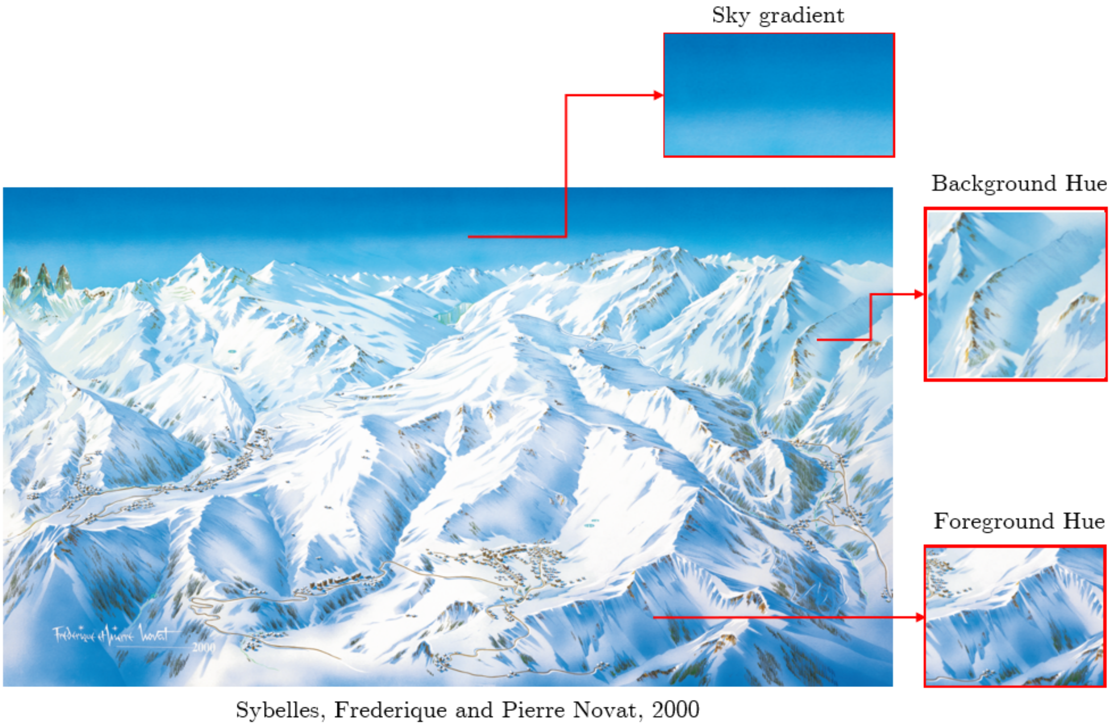


Figure 3.17: Diagram showcasing blending condition for final intensity-map

The sky gradient is fairly simple to simulate, it basically consists of a quad texture over which, the required gradient of blue is filled in. When rendering the final terrain in 3-D, the sky-quad is placed behind the terrain; as the terrain is in the foreground, in front of the sky-quad, fragments belonging to the terrain are rendered first (depth-writes are disabled by setting glDepthMask(GL_FALSE)).

In order to assign the intensity values to a specific color, choosing a color palette specific to the panorama maps is essential. The interface developed allows the user to select a color palette for the shadowed as well as lit regions. The selected colors are then used to form a transfer function (mapping intensity to final hue output), aptly referred to as *attribute maps* [BTM06].

Attribute maps, originally coined by [BTM06]- can be explained through the Figure 3.18 below; It is basically a 2-D texture that has varying properties on both axes. Along the horizontal, there is a variation in palette (discrete : toon texture or continuous), which is then mapped to the Lambertian shading value ($\overrightarrow{n} \cdot \overrightarrow{l}$, n : surface normal and l : unit light direction) at that point. The vertical axis is responsible for tone detail that is dependent on an attribute (orientation or depth) "D". Ideally, this attribute-map texture can be viewed as a stack of 1-D tone textures.

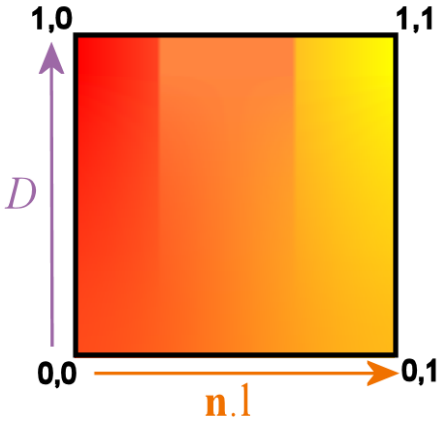


Figure 3.18: Attribute map as defined in the paper by [BTM06]

The thesis differs slightly in the creation of these attribute maps and proposes a subset termed as : *Aerial Perspective Attribute Maps* (APAMs) (see Figure 3.19). Several points can be noted:

- APAMs differ from the attribute maps presented earlier along both axes. In the sense that, the pair $(intensity, opticaldepth)$ is mapped to a tone value that corresponds to a dark to light transition along the X-axis. Along the vertical, as the name for APAM suggests, the variation in aerial perspective is mapped to an increasing tint from the base tone.

- For creating APAMs, the color palette that is chosen by the user is taken as the base value on the map. APAMs can be effectively viewed as a stack of 1-D tone-textures;

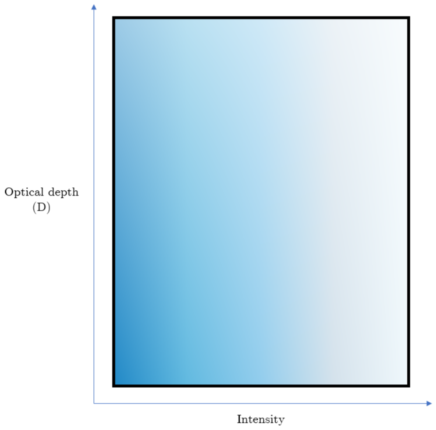


Figure 3.19: A sample APAM for mapping $(intensity, opticaldepth)$ to the selected color palette

as the base 1-D texture is specified, vertical tone variation is created through interpolation with the base tone using Equation 3.4. An additional point to note is that values along both axes are continuous as opposed to variations specified in [BTM06].

- Aerial perspective is defined as the phenomenon wherein the saturation and contrast of an object tends to decrease with increasing distance from the viewing plane. Therefore, objects that appear far away tend to have a dull bluish-hue as opposed to objects present in the foreground. The Novat panoramas showcase this effect in a less-pronounced manner; although the saturation decreases, mountains in the background still have distinct boundaries with muted shading (as shown in Figure 3.20).

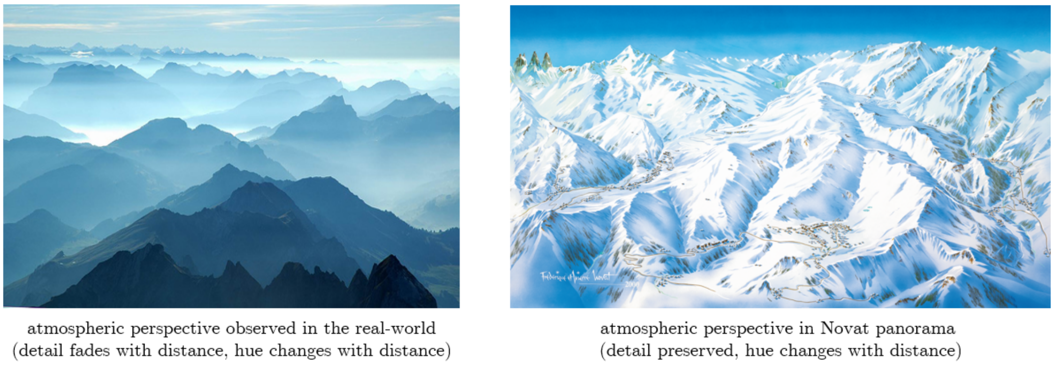

atmospheric perspective observed in the real-world
(detail fades with distance, hue changes with distance)
atmospheric perspective in Novat panorama
(detail preserved, hue changes with distance)

Figure 3.20: Differences between real-world aerial perspective (left) and Novat panoramas (right)

- In order to compensate for this change in hue due to aerial perspective, the tone is made lighter with increasing distance, given by the relation :

$$Color \rightarrow lerp(tone_y, tone_y + (tint \cdot (1 - tone_y))) \tag{3.4}$$

tint is a controllable tone-tint factor/parameter that allows the user to make the tone-variation along Y lighter or closer to the base tone, depending on the amount of aerial perspective effect needed.

- The optical depth is given as follows:

$$opticalDepth \rightarrow clamp(e^{-dist(viewPosition, terrainPosition)}, 0, 1) \tag{3.5}$$

Therefore, points closer to the viewing plane will have a color that is mapped to positions around the base-tone present at the bottom of the APAM texture.

- As a final point, the pipeline allows the user to create APAMs (both for shadowed and lit regions) that appear as the one shown in Figure 3.19, depending on the color palette chosen and tint-factor set. However, needless to say, attribute-map textures can also be created externally and loaded into the pipeline.

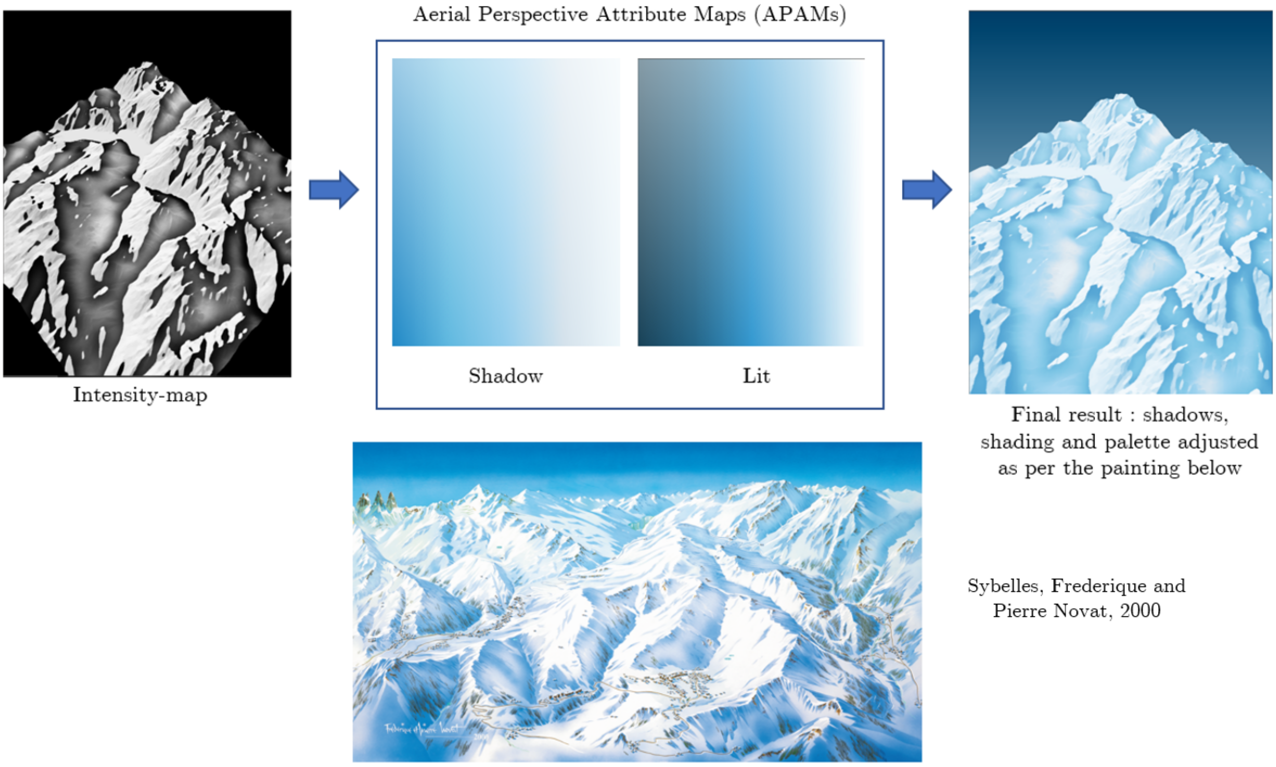
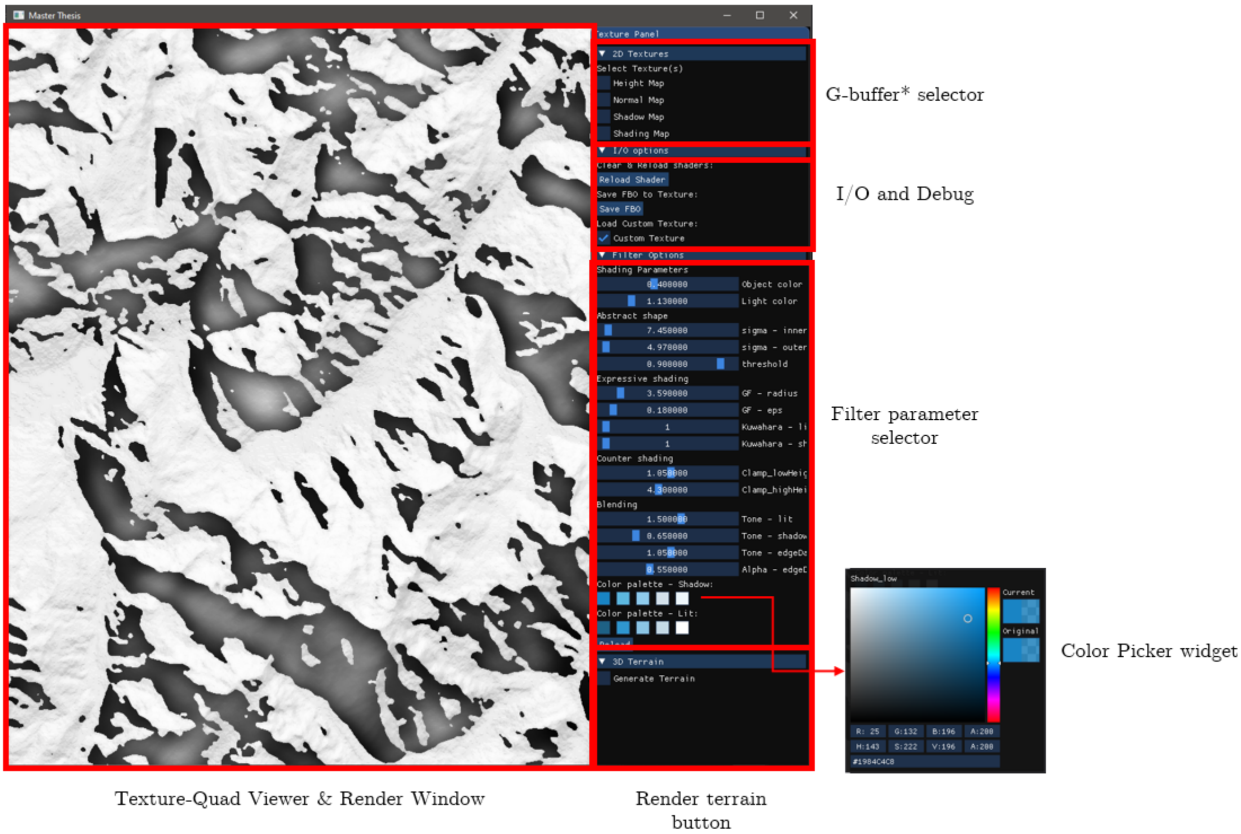

Aerial Perspective Attribute Maps (APAMs)
Intensity-map
Shadow
Lit
Final result : shadows, shading and palette adjusted as per the painting below
Sybelles, Frederique and Pierre Novat, 2000

Figure 3.21: Diagram showcasing the final colored result in accordance with one of Pierre Novat's paintings.

3.2 Implementation & parameter tuning

The rendering framework has been implemented in C++14 and ImGui [4]. The rendering itself is done in OpenGL 4.3 and GLSL 4.5. As mentioned prior, most of the methods involving texture-processing are calculated on the GPU through fragment shaders. Two passes are considered; the first pass calculates shadows, normals, diffuse shading and stores them as usable textures (pass occurs only once). In the second pass (iterative), the stored textures are used through the pipeline shown in figure 3.1.

The renderer includes a main-texture(quad) viewer with a side panel for configurable parameters (as shown in figure). Additionally, a camera, light and mesh are also generated. The DEM data used belongs to the region : Alpe d'Huez, that consists of 800 x 1200 map with a pitch of 25m (span of 24km^2 resulting in a mesh of 5748006 triangles. The framework has been developed using the GPU : GTX 1060 (laptop) and runs at a stable 60 frames per second. Frame-rate drops that occur are variable and depend on changes done by the user (increasing kernel size, adding additional parameters), to the filters at various stages of the pipeline.



Texture Panel

▼ 2D Textures

Select Texture(s)

□ Height Map
□ Normal Map
□ Shadow Map
□ Shading Map

▼ I/O options

Clear & Reload shaders:

Reload Shader

Save FBO to Texture:

Save FBO

Load Custom Texture:

☑ Custom Texture

▼ Filter Options

Shading Parameters

0.400000	Object color
1.130000	Light color

Abstract shape

7.450000	sigma - inner
4.970000	sigma - outer
0.900000	threshold

Expressive shading

3.590000	GF - radius
0.180000	GF - eps
1	Kuwahara - li
1	Kuwahara - sh

Counter shading

1.050000	Clamp_lowHeig
4.300000	Clamp_highHei

Blending

1.500000	Tone - lit
0.650000	Tone - shadow
1.050000	Tone - edgeDa
0.550000	Alpha - edgeD

Color palette - Shadow:

Color palette - Lit:

Reload

▼ 3D Terrain

□ Generate Terrain

Shadow_low

Current

Original

R: 25	G:132	B:196	A:200
H:143	S:222	V:196	A:200

#1984C4C8

Figure 3.22: Figure showcasing main application window with sub-windows having different options

[4] https://github.com/ocornut/imgui

The application can be divided into five main parts:

- **Render Window** : This area mainly displays two forms of output; first, the resulting texture quad from filtering operations. Second, the final 3-D rendered terrain with the texture mapped on top. When displaying the 3-D terrain, the user is given options for navigation (W,A,S,D keys for scene navigation with mouse for rotation).
- **G-buffer selector** : This area allows selection and viewing of four primary textures - height, normal, shadow and shading maps.
- **I/O and debug** : Contains buttons for debugging changes in the fragment shader, saving current framebuffer (texture that is displayed) to memory and loading external textures.
- **Filter parameters** : Contains sliders for parameters detailed within algorithms detailed prior. Color palette consists of 5 primary colors for both shadowed and lit regions, which can be selected in a separate widget window. Note, since the implementation does not allow real-time changes, a button ("reload") needs to be clicked for changes to take effect.
  - *Shading parameters* : Consist of object and light colors that can be modified. Usually set between (0.25 - 0.5) for object and (0.9 - 1.2) for light respectively.
  - *Abstract shape* : Used for modifying shape of the shadow. Consists of blur kernel radius and threshold (default to kernel size of 7 and threshold of 0.9). Increasing kernel size, increases abstraction. Threshold determines inclusion of binary regions.
  - *Expressive shading* : Consists of guided filter (kernel radius and regularization parameter) and kuwahara filter (lit and shadowed region kernel sizes). Variation in parameters is explained in Chapter 4.
  - *Counter shading* : Consists of two clamps - low and high, that are used to vary the edge-thickness of the shadow.
  - *Blending* : Consists of four parameters that are used to change tone values for shading, inter-reflection and edge-darkened maps along with $alpha$ that is used to control intensity for the combined stylized shading. Tone-parameter values, if set greater than 1, darken the map, while less than 1 lightens it (tone for edge-dark $<$ tone shading-shadowed region). On the other hand, increasing $alpha$, darkens the map and brightens it when decreased. Depending on the type of texture and tone needed, the parameters can be suitably adjusted.
- **Render terrain** : Maps current texture to the terrain and renders it within the main window.

4 Discussion & results

This chapter will mainly delve deeper into some of the core methods introduced in the previous chapter. At the end, a couple of results along with the corresponding panorama map art-work will be showcased.

4.1 Stylized shading

The chapter begins with a deeper look into the shading method introduced in the previous chapter (see Figure 4.1). To reiterate, a novel shading method is proposed that aims to simulate the shading used in the panorama maps by mimicking the artistic process of Novat (See Chapter 3). In this method, there is a heavy emphasis given to edge-preserving smoothing filters, mainly the guided filter [HST10] and the kuwahara filter [Kuw+76]. This section mainly serves as a continuation from the stylized shading subsection in Chapter 3. The Edge-preserving smoothing filters detailed, aim to mimic the watercolor pencil outlines (including smoothing) and gouache overlay.

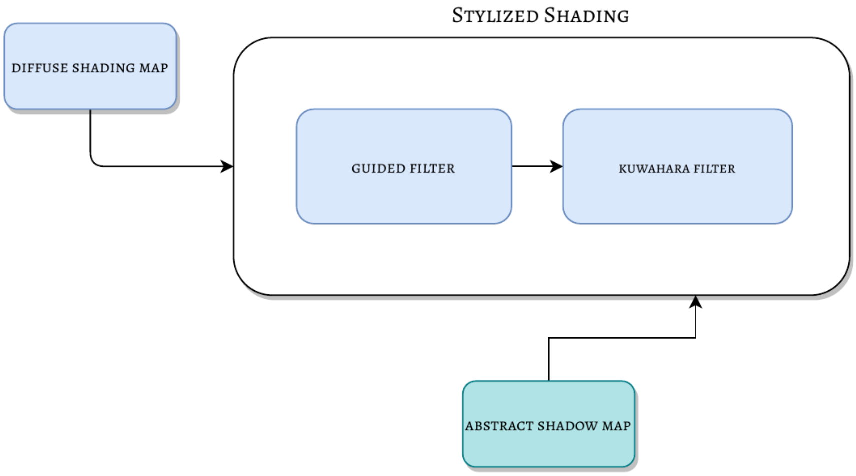


Figure 4.1: Stylized shading sub-steps

The guided filter is a fairly recent (2010) image filtering technique that is derived from a local linear model. The unique aspect of this filter comes from the fact that is generates a filter-output by considering a *guidance image* as input. This guidance image can be the original input image itself or some other image. Depending on the input image used, multiple image-based operations such as noise-reduction, edge-preserved smoothing, HDR compression, image matting/ feathering, haze removal etc. can be done. More recently, in real-time rendering applications (SSAO, screen-space reflections etc.), guided filters have been used as effective de-noisers. Additionally, they also seem to have a significant advantage over joint-bilateral filters.

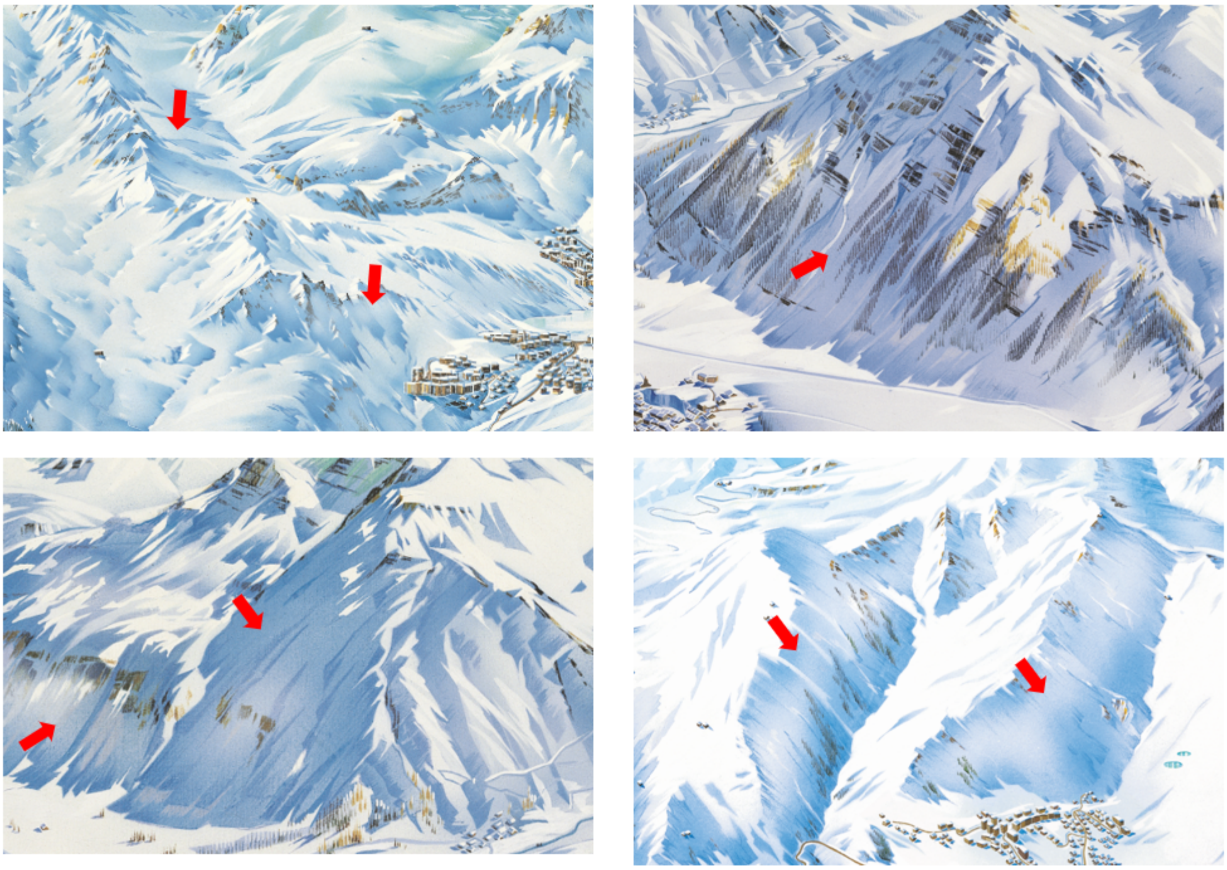
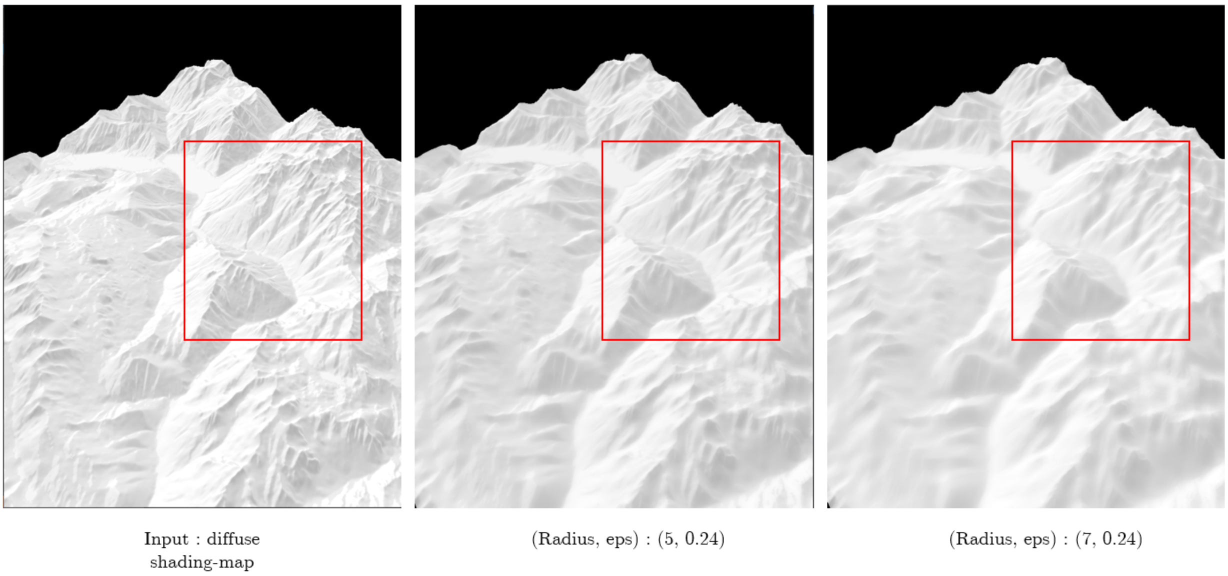


Figure 4.2: Figure showcasing differences in shading within the shadowed region (red arrows indicate differences in stroke variation) among different Novat panoramas.

In the context of this thesis and the use-case of panorama maps, the guided filter can be used to replicate the artistic effect of edge-emphasis and smooth-gradients in watercolor pencils. To give a brief description, the main idea behind the algorithm is simple : given a number of local samples, the goal is to find the *best-fit* linear relationship between the guidance and filtered signal. This is effectively similar to a linear-regression problem, wherein, pixels within the kernel are checked for high (edges) /low (patches) frequency components with respect to the pixels present in the guidance image. Algorithm 2 gives an overview of the guided filter. Additionally, the filter tends to infer the scaling parameter from the data, should the image operate in multiple scales.

In the Figure 4.3 above, there are two parameters that can be controlled, $radius$: radius of the filter-kernel and ϵ : regularization parameter that controls smoothness . Upon increasing the radius, it can be observed that although the image becomes smoother, there is no significant change to the edge-profiles. Advantages of guided filters can be given as follows :

- Really simple and efficient to compute (with a time complexity of O(N)).



Input : diffuse shading-map

(Radius, eps) : (5, 0.24)

(Radius, eps) : (7, 0.24)

Figure 4.3: Guided filtering showcasing edge-preserved smoothing, increase in radius increases smoothing but preserves edge-profiles

- Works at multiple image-scales; at it's core, the filter is basically performs linear regression, which in turn can be adapted to work at multiple scales.
- Edges (and smaller details) are preserved better and images tend to look cleaner overall.
- No piecewise-constant artifacts : chunkiness, gradient-reversals etc. are eliminated resulting in a more natural, less processed image.

The guided filter, however, is not without faults and has a couple of problem. The disadvantages can be given as follows:

- "Smudging" : If the regression line does not fit the input signal well, this might create discontinuities around edges.
- Similar to the above point, the guided filter entirely depends on it's ability to best-fit points. Failure to do so may lead to the creation of janky artifacts.
- Multi-variable regression can be considered as an alternative to linear regression as it allows for better edge-preservation, better smoothing and lower artifacts. But it is far too costly and may not be applicable in the case of multiple channels.
- Increasing bit-depth for greater precision increases memory requirements.

An alternative to guided filter that was initially explored, was the joint-bilateral filter [ED04; Pet+04]. The bilateral filter when applied on a pixel, basically computes the weighted average of neighboring pixels. Another variant of the bilateral filter called the joint bilateral filter behaves similarly to the guided filter; in the sense that it uses a "guide" signal to create weights for filtering another signal. (Figure 4.4 shows the comparison between a bilateral filter and guided filter having the same kernel radius). From the image, it can be noted that the guided filter produces sharper, more distinct edges while retaining the smoothness around the edge-neighborhood.

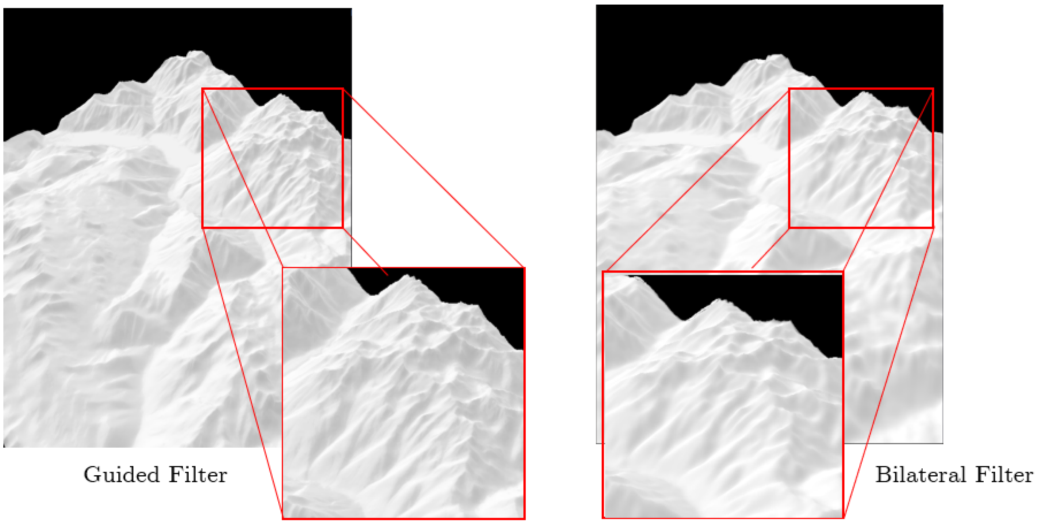
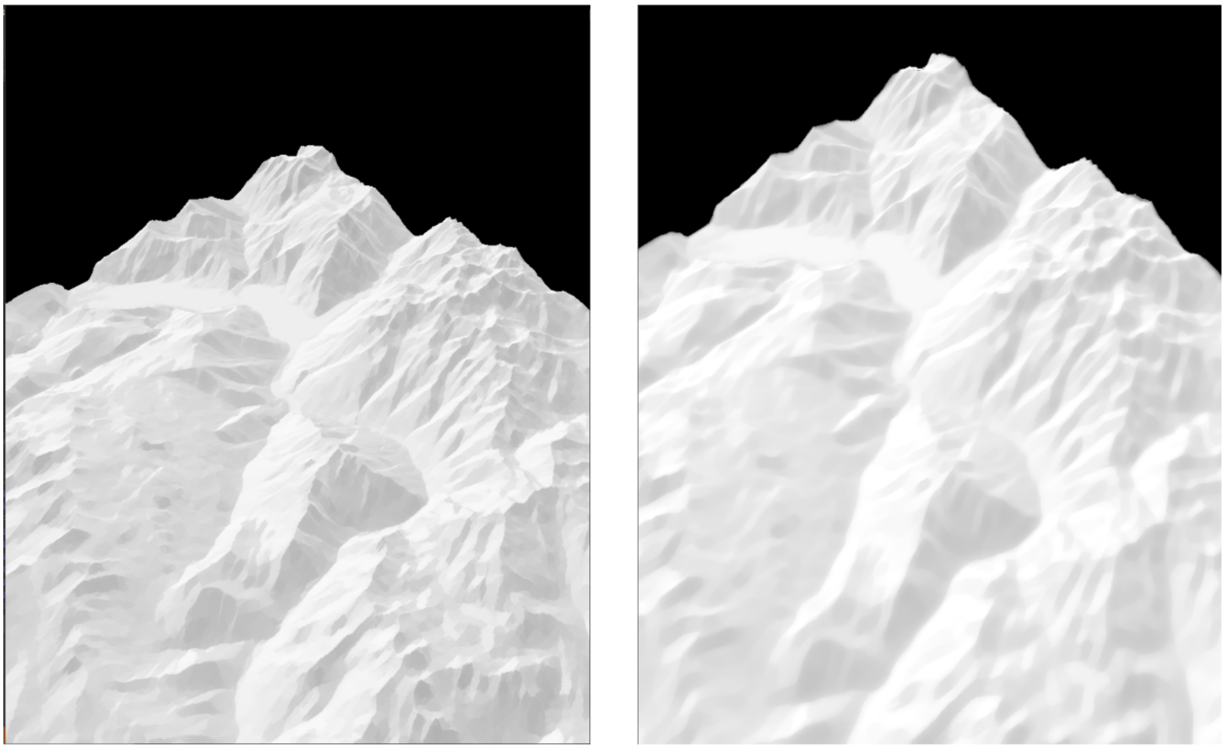


Figure 4.4: Comparison between guided and bilateral filter

In the second step of the stylized shading pipeline, the kuwahara filter [Kuw+76] is applied. This is mainly to simulate brush strokes along with acrylic usage on top a transparent watercolor layer. As shading differs from shadowed to lit region, the kernel size is adjusted (user-controllable) to allow stroke-variation (shadowed areas are smoother, with less detail which leads to a smaller kernel size and vice-versa for lit regions). Figure 4.5 further reinforces the choice of guided filter over bilateral filter for achieving a painterly Novat effect visible in the panorama maps.



Guided filter(5,0.24) + Kuwahara

Bilateral filter(5) + Kuwahara

Figure 4.5: Comparison between guided and bilateral filter after the iterative application of kuwahara filter

4.2 Shadow stylization

This section will be relatively short and mainly focus on changes observed in shadow abstraction along with the edge-darkening effect. Shadow abstraction is an important step in our rendering pipeline and within the shadow stylization process. By changing the kernel size and applying a threshold, the level of detail of shadows can be controlled.

Drastic increase or decrease in kernel size may lead to shadows that cause a shift in the perception of the scene. In the case of Novat's artwork, most cases lie in-between the first and second figures (within Figure 4.7 and rarely exceed the second figure). However, at times, the user may want to create local edits to shadows. For example, (s)he may want to extend the boundary over some valley or change the level of detail on a shadow of a singular mountain face.

Unfortunately, this forms one of the major limitations to filter-based approaches; most user-controllable parameters that have been used in the proposed framework to create global changes. However, this is an interesting avenue for future work that may allow better control in modifying shadows.

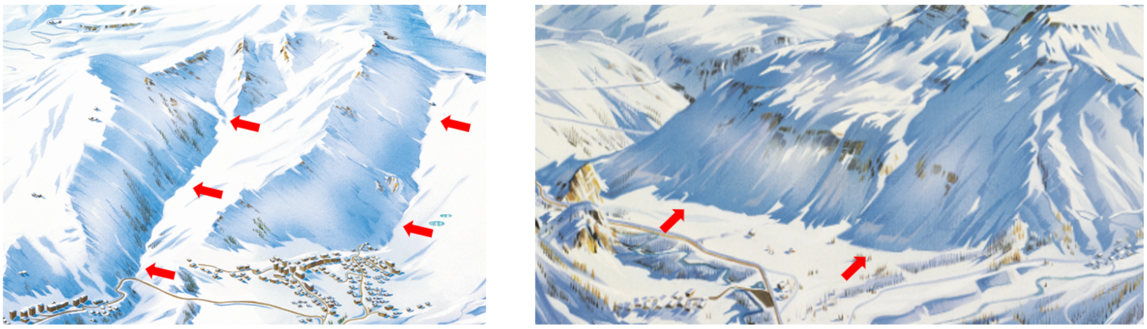


Figure 4.6: Figure showcasing hard, well-defined boundaries yet fairly abstract with a neat -cut at the valley (red arrows follow curvature of the shadow).

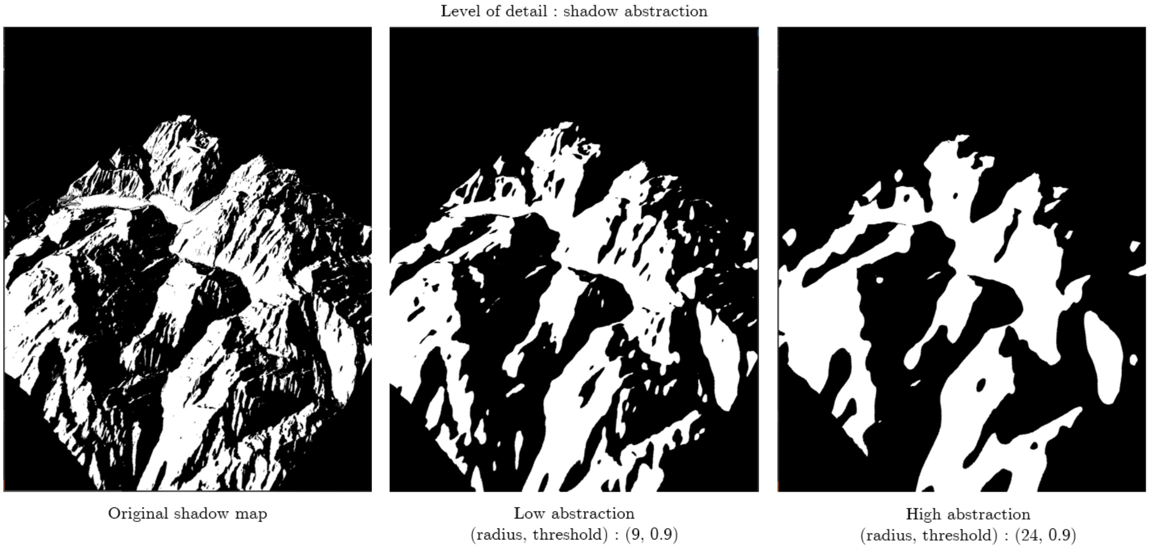
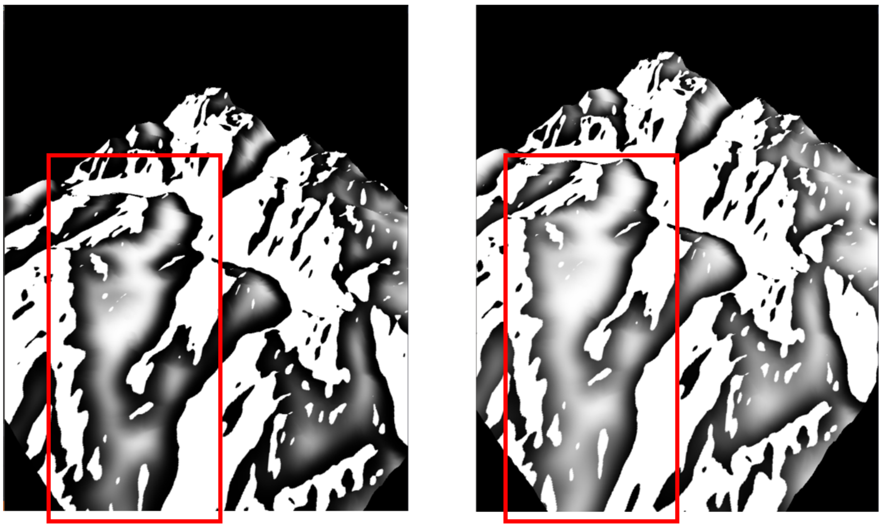

Level of detail : shadow abstraction
Original shadow map
Low abstraction
(radius, threshold) : (9, 0.9)
High abstraction
(radius, threshold) : (24, 0.9)

Figure 4.7: Increasing level of abstraction : user controllable through Gaussian blur kernel radius and threshold

In the case of edge-darkened shadows, there are multiple paintings wherein the edges tend to only be present on the bottom side of the mountain face. Other cases include darkened edges on both (top and bottom) sides of the mountain face. User-controllable parameters (explained in Chapter 3) allow some amount of flexibility in controlling the variation of edge-thickness with the height.



Edge-darkening from both sides of mountain face

Edge-darkening from bottom of mountain face

Figure 4.8: Changes in edge-thickness : darkening both sides (top and bottom) of mountain face v/s bottom of mountain face

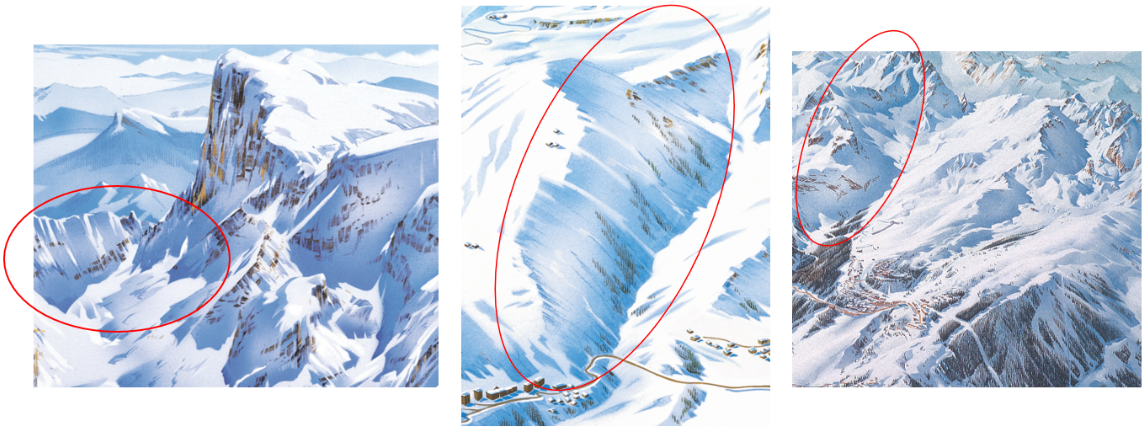
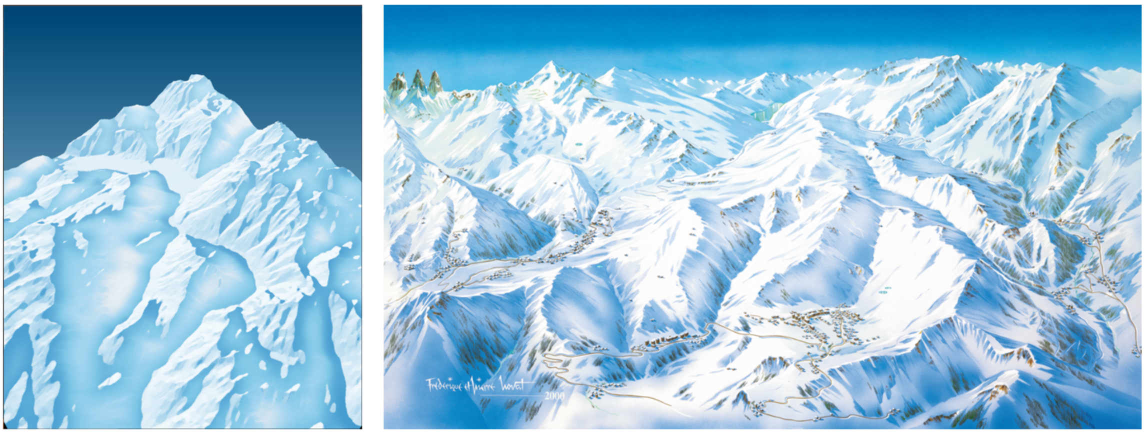


Figure 4.9: Figure showcasing the edge darkening effect across multiple panoramas (visible in regions within red oval, with varying contrast difference between central reflections and gradient towards the edge)

4.3 Hue calculation : results

This section will mainly discuss the final result of mapping the intensity-map to a certain hue. A point to note : the viewpoint is fixed for the first two results below, the emphasis will be on changes to the art-style. Two cases are considered with different artists yet following the same school of art (Atelier Novat), these are discussed below:

- The first case considered is the painting of Sybelles by Frederique and Pierre Novat in 2000. Upon careful observation, there are several distinct features visible (in the context of shadows) : shadows are fairly abstract and have smooth, well-defined boundaries, edges of the shadows are fairly dark to showcase a difference in contrast, underlying shading in the shadowed area is fairly smooth.

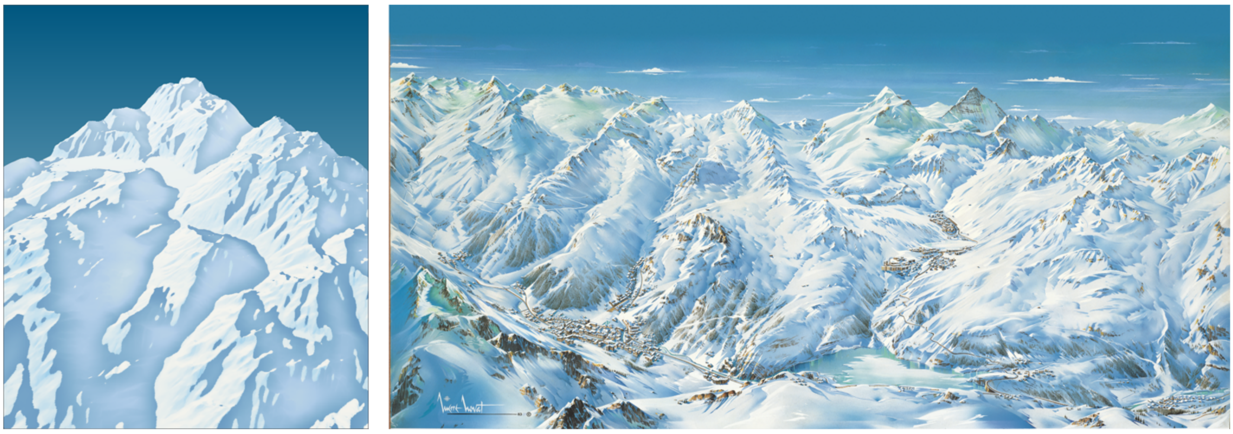


Result (case 1)

Sybelles, Frederique and Pierre Novat, 2000

Figure 4.10: Hue result : case 1 (left), Sybelles, Frederique and Pierre Novat, 2000 (right)

- The second case considered is the painting of Espace Killy by Pierre Novat in 1983. As opposed to the previous result, this one is more slightly more subdued; colors are muted and the painting is made to appear far more natural. With regards to shadows, the effects such as edge-darkening and inter-reflections are less noticeable. Additionally, within the shadow, there are fewer gradient levels visible. Edge-boundaries are less abstract and more in-line with how they should appear realistically (although some abstraction remains).



Result (case 2)

Espace Killy, Pierre Novat, 1983

Figure 4.11: Hue result : case 2 (left), Espace Killy, Pierre Novat, 1983 (right)

- Apart from using standard DEMs, the pipeline can also be used for open-sourced height-maps, derived from OpenStreetMap or the USGS shaded relief. An example height-map of the Chamonix valley region (50 sq.km) is taken [1] and filter parameters are tweaked to allow a style closer to case 1 above (figure 4.10).

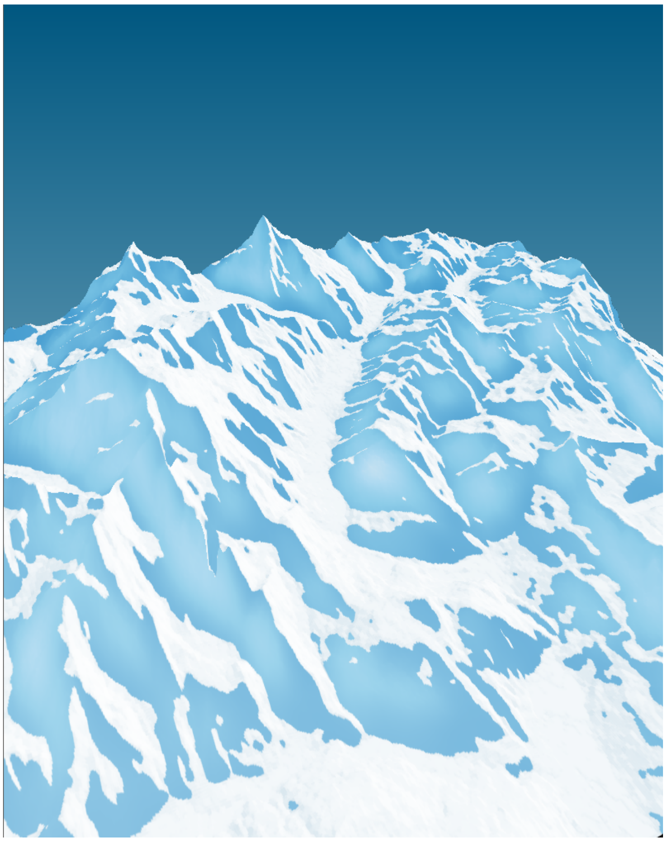


Figure 4.12: Hue result : Chamonix valley region including Mont Blanc

[1] height-map extractor : http://terrain.party/

5 Conclusion & future work

On an ending note, this chapter will focus on future work and will conclude the thesis by giving a brief recap of the methods proposed prior. With that said, there is still quite a lot of work that needs to be done in order to get a true rendering of the Novat-style panorama painting. Following are a few ideas that could be improved upon :

- The shading proposed in this thesis is not perfect by any means despite achieving a painterly-look that aligns with shading done in the panorama maps. In fact, there are several limitations such as appearance of unwanted artifacts, smudging, no real-time implementation (to name a few). However, edge-preserving, smoothing filters seem to mimic the artistic process pretty well and should be explored further.
- In terms of the shadow-stylization section of the pipeline, most effects undergo global changes when a certain set of parameters is tweaked. For example, in shadow abstraction and edge-darkening, changing the filter-kernel size results in changes being reflected throughout the texture. Therefore, methods to include local editing should be looked into.
- As mentioned in the previous point, the current proposed implementation relies on parameter-based stylization to showcase effects. Although simple to implement and effective, it may not be the correct approach in terms of user-intuitiveness. In most cases, the user should be given a higher amount of creative control with real-time feedback.
- In the same vein, there have been recent advances in neural style transfer methods, that allow for transferring artistic style from one image to another. This could be used to set a base-line guide for the user to allow a better understanding of the Novat style.
- There are several decorative elements such as rocks, trees, houses etc. that need to be addressed as well. Each one of these elements have a specific ruleset for positioning, shading and shadows withing the panorama painting.
- Finally, the current implementation should be made to work in conjunction with the *terrain deformation* framework developed at the LIG (Laboratoire d'Informatique de Grenoble), to ensure a sense of completeness for simulating the Novat-style (Terrain deformation plays a central role in Novat's panorama construction).

To conclude, the thesis proposes two contributions; a brief study into the artistic style of Atelier Novat along with a rendering framework that takes a set of DEM files as input and produces an image that resembles a ski-panorama map drawn in the style of Atelier Novat. Although there are a myriad of elements that make up a panorama map, the thesis addressed primarily cast shadows and methods for shadow stylization.

The pipeline deals with three components : a novel method for stylized shading, a set of methods for modifying structural properties of cast shadows (according to an artist-driven ruleset) and, finally, a transfer function to map the combined, shadow-intensity texture to a color-palette resembling the panorama maps. Consequently, a major portion of the framework is based on image-space filters that are run on the GPU through fragment shaders. The latter supports interactive image generation, which facilitates the setting of parameters and choosing of the viewpoint.

A Appendix : Panorama maps by Atelier Novat

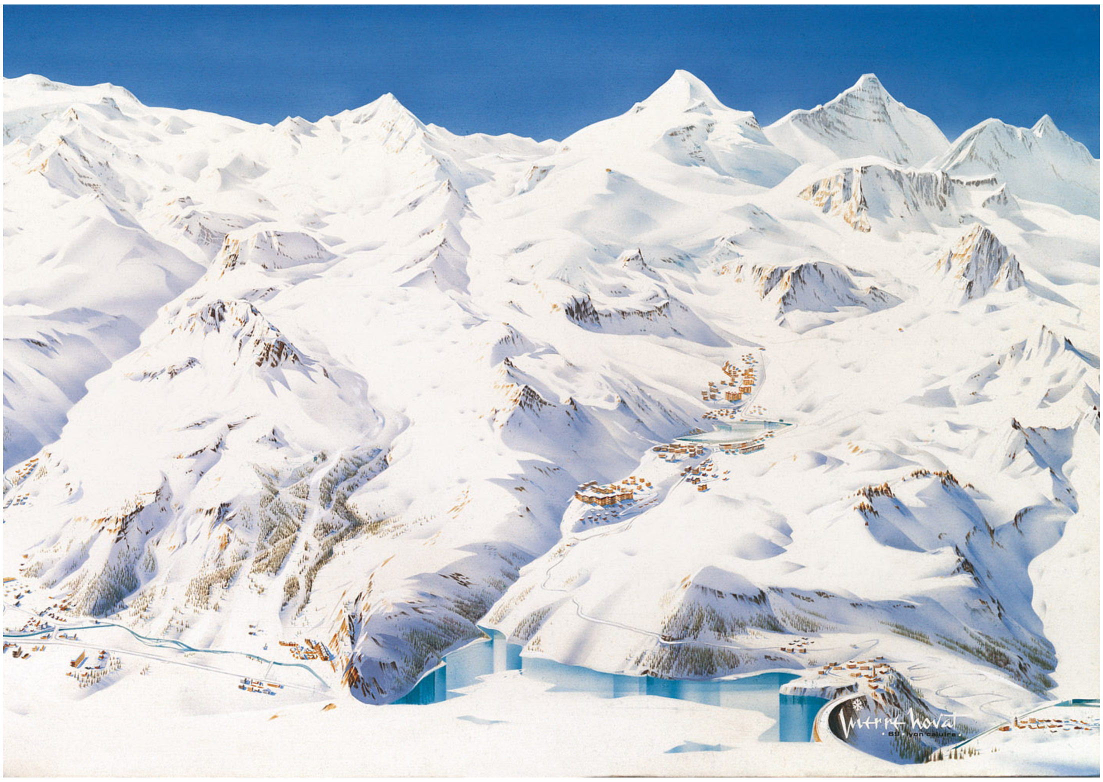


Figure A.1: Val d'Isere, Tignes, Pierre Novat, 1967.

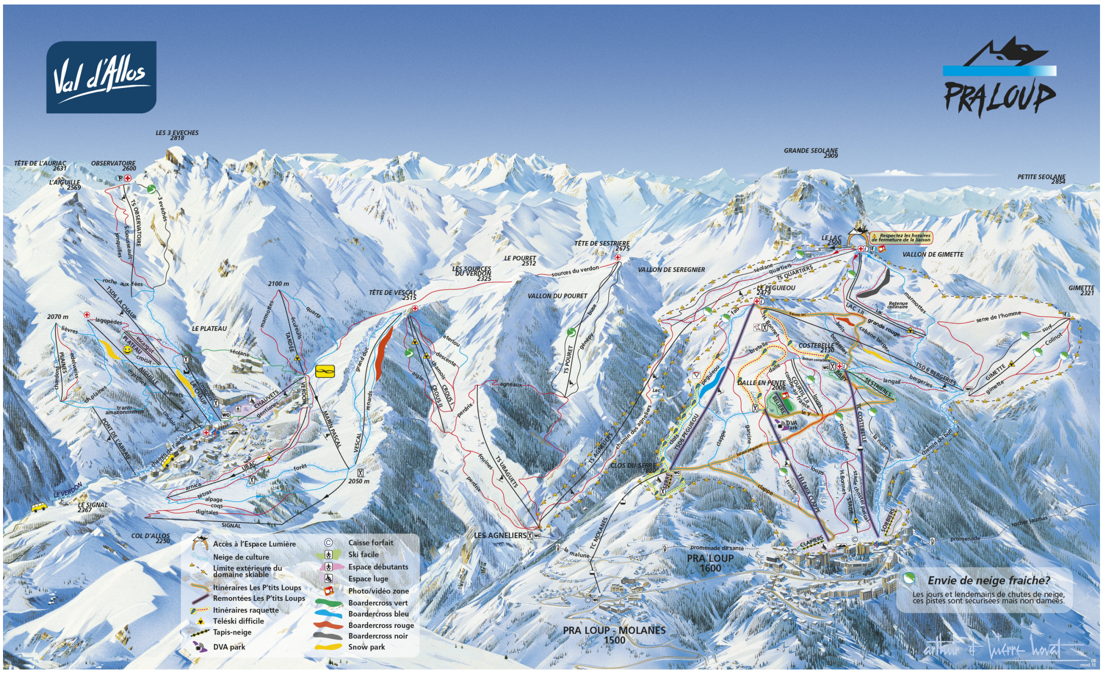


Figure A.2: Praloup, Arthur and Pierre Novat

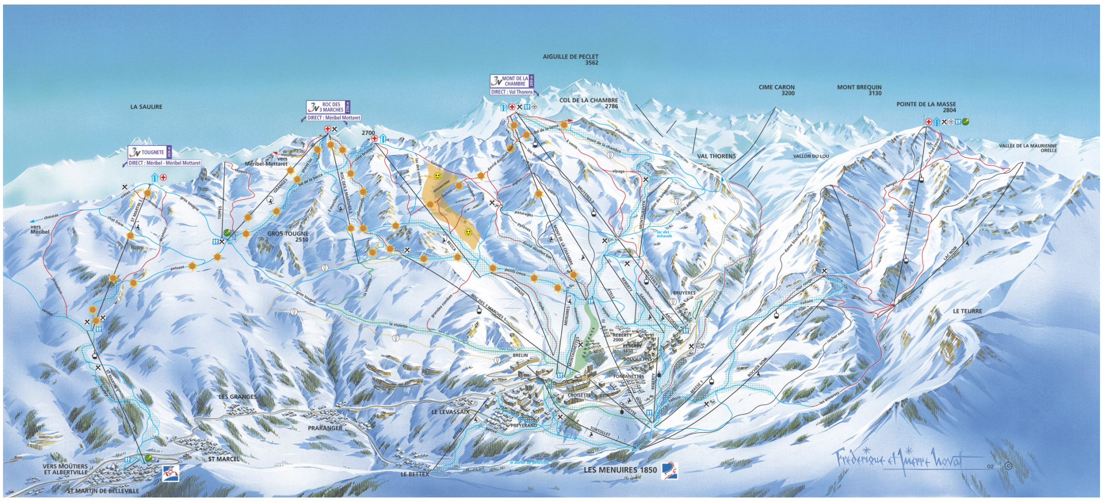


Figure A.3: Les 3 Vallées (Val Thorens, Les Menuires, St Martin De Belleville, Meribel, Brides-les- bains, La Tania, Courchevel) , Frédérique and Pierre Novat, 2007

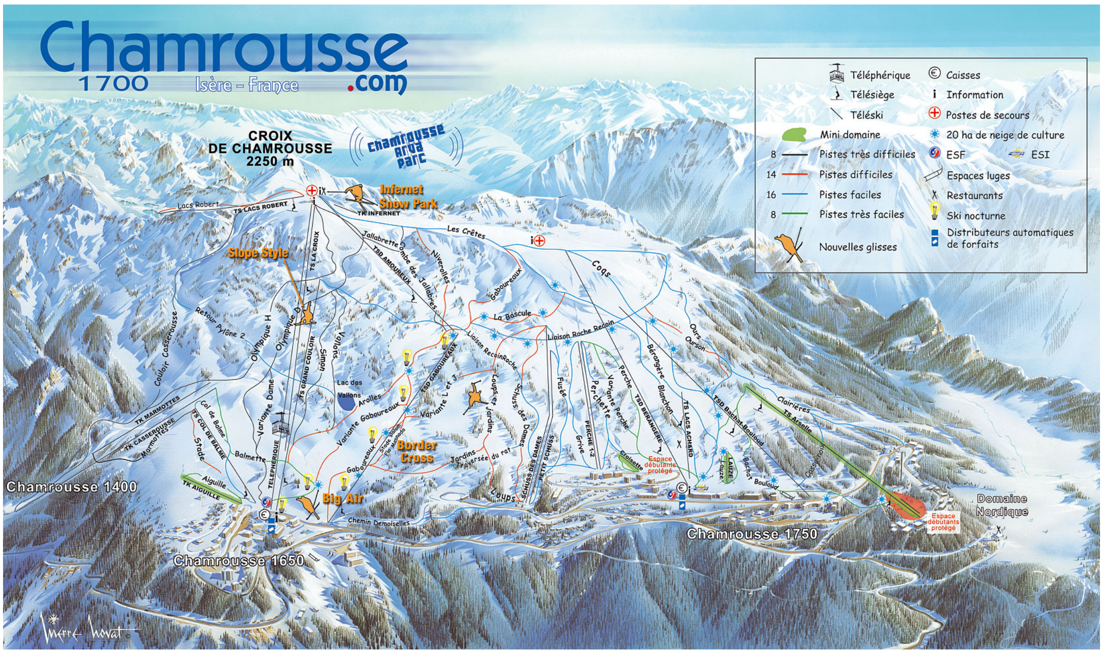


Figure A.4: Chamrousse, Pierre Novat, 2005

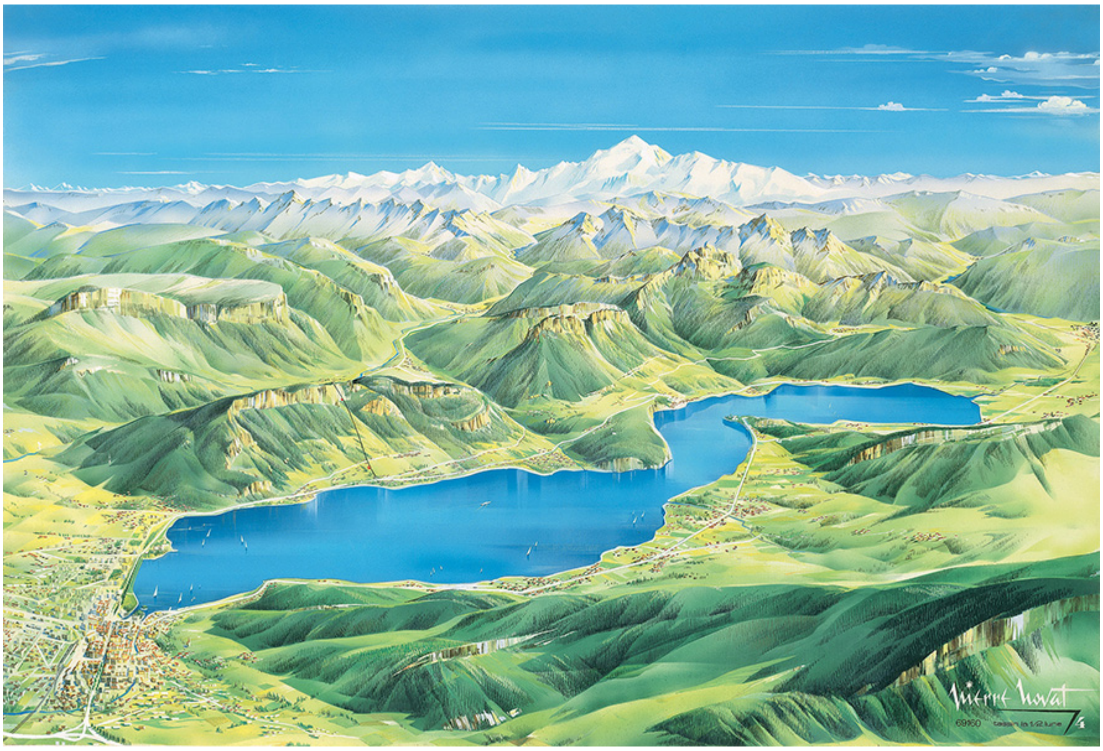


Figure A.5: Lac d’annecy en été, Pierre Novat

Bibliography

[Bar16] Krzysztof Bartyzel. “Adaptive kuwahara filter”. In: *Signal, Image and Video Processing* 10.4 (2016), pp. 663–670.

[Bou+06] Adrien Bousseau et al. “Interactive watercolor rendering with temporal coherence and abstraction”. In: *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*. ACM. 2006, pp. 141–149.

[BS17] S Alex Brown and Faramarz Samavati. “Real-time panorama maps”. In: *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. ACM. 2017, p. 6.

[BST09] Margarita Bratkova, Peter Shirley, and William B Thompson. “Artistic rendering of mountainous terrain.” In: *ACM Trans. Graph.* 28.4 (2009), pp. 102–1.

[BTM06] Pascal Barla, Joëlle Thollot, and Lee Markosian. “X-toon: an extended toon shader”. In: *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*. 2006, pp. 127–132.

[CM02] Dorin Comaniciu and Peter Meer. “Mean shift: A robust approach toward feature space analysis”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5 (2002), pp. 603–619.

[Cur+97] Cassidy Curtis et al. “Computer-Generated Watercolor”. In: *Proc. SIGGRAPH1997* 97 (June 1997). DOI: 10.1145/258734.258896.

[DeC+07] Christopher DeCoro et al. “Stylized shadows”. In: *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*. 2007, pp. 77–83.

[DS02] Doug DeCarlo and Anthony Santella. “Stylization and abstraction of photographs”. In: *ACM transactions on graphics (TOG)* 21.3 (2002), pp. 769–776.

[DVP19] François Desrichard, David Vanderhaeghe, and Mathias Paulin. “Global Illumination Shadow Layers”. In: *Computer Graphics Forum*. Vol. 38. 4. Wiley Online Library. 2019, pp. 183–191.

[ED04] Elmar Eisemann and Frédo Durand. “Flash Photography Enhancement via Intrinsic Relighting”. In: *ACM Transactions on Graphics (Proceedings of Siggraph Conference)* 23 (Aug. 2004). DOI: 10.1145/1015706.1015778.

[Eis+11] Elmar Eisemann et al. *Real-time shadows*. CRC Press, 2011.

[EKD08] Jan Eric Kyprianidis and Jürgen Döllner. “Image Abstraction by Structure Adaptive Filtering.” In: Jan. 2008, pp. 51–58. DOI: 10.2312/LocalChapterEvents/TPCG/TPCG08/051-058.

[Gao97] Jay Gao. “Resolution and accuracy of terrain representation by grid DEMs at a micro-scale”. In: *International Journal of Geographical Information Science* 11.2 (1997), pp. 199–212.

[GDG19] Junjie Gao, li Dajin, and Wenran Gao. “Oil Painting Style Rendering based on Kuwahara Filter”. In: *IEEE Access* PP (July 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2931037.

[Has+03] Jean-Marc Hasenfratz et al. “A survey of real-time soft shadows algorithms”. In: (2003).

[Her+01] Aaron Hertzmann et al. “Image analogies”. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001, pp. 327–340.

[HST10] Kaiming He, Jian Sun, and Xiaoou Tang. “Guided image filtering”. In: *European conference on computer vision*. Springer. 2010, pp. 1–14.

[JD78] Frederick J. Doyle. “DIGITAL TERRAIN MODELS: AN OVERVIEW.” In: *Photogrammetric Engineering and Remote Sensing* 44 (Dec. 1978), pp. 1481–1485.

[Kaj86] James T Kajiya. “The rendering equation”. In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 1986, pp. 143–150.

[KKD09] Jan Eric Kyprianidis, Henry Kang, and Jürgen Döllner. “Image and video abstraction by anisotropic Kuwahara filtering”. In: *Computer Graphics Forum*. Vol. 28. 7. Wiley Online Library. 2009, pp. 1955–1963.

[KL08] Henry Kang and Seungyong Lee. “Shape-simplifying image abstraction”. In: *Computer Graphics Forum*. Vol. 27. 7. Wiley Online Library. 2008, pp. 1773–1780.

[KP98] Karl Kraus and Norbert Pfeifer. “Determination of terrain models in wooded areas with airborne laser scanner data”. In: *ISPRS Journal of Photogrammetry and remote Sensing* 53.4 (1998), pp. 193–203.

[Kuw+76] MKSMK Kuwahara et al. “Processing of RI-angiocardiographic images”. In: *Digital processing of biomedical images*. Springer, 1976, pp. 187–202.

[LD05] Thomas Luft and Oliver Deussen. “Interactive watercolor animations”. In: *Computer Graphics and Applications (PG’05)*. 2005.

[Liu08] Xiaoye Liu. “Airborne LiDAR for DEM generation: some critical issues”. In: *Progress in Physical Geography* 32.1 (2008), pp. 31–49.

[Mil57] Charles Leslie Miller. *The Spatial Model Concept of Photogrammetry*. MIT Photogrammetry Laboratory, 1957.

[Mil86] Gavin SP Miller. “The definition and rendering of terrain maps”. In: *ACM SIGGRAPH Computer Graphics*. Vol. 20. 4. ACM. 1986, pp. 39–48.

[MIW13] Oliver Mattausch, Takeo Igarashi, and Michael Wimmer. “Freeform shadow boundary editing”. In: *Computer Graphics Forum*. Vol. 32. 2pt2. Wiley Online Library. 2013, pp. 175–184.

[Mus] F Kenton Musgrave. “FractalWorlds. com musgrave@ fractalworlds. com www. fractalworlds. com/musgrave”. In: ().

[Pat00] Tom Patterson. “A view from on high: Heinrich Berann’s panoramas and landscape visualization techniques for the US National Park Service”. In: *Cartographic Perspectives* 36 (2000), pp. 38–65.

[PD06] Sylvain Paris and Frédo Durand. "A fast approximation of the bilateral filter using a signal processing approach". In: *European conference on computer vision*. Springer. 2006, pp. 568–580.

[Per85] Ken Perlin. "An image synthesizer". In: *ACM Siggraph Computer Graphics* 19.3 (1985), pp. 287–296.

[Pet+04] Georg Petschnigg et al. "Digital photography with flash and no-flash image pairs". In: *ACM transactions on graphics (TOG)* 23.3 (2004), pp. 664–672.

[Ram06] J Raul Ramirez. "A new approach to relief representation". In: *Surveying and Land Information Science* 66.1 (2006), pp. 19–25.

[Rit+10] Tobias Ritschel et al. "Interactive on-surface signal deformation". In: *ACM SIGGRAPH 2010 papers*. 2010, pp. 1–8.

[War82] Stephen G Warren. "Optical properties of snow". In: *Reviews of Geophysics* 20.1 (1982), pp. 67–89.

[Wei] Joachim Weickert. *Anisotropic diffusion in image processing*. Vol. 1.

[WPF90] Andrew Woo, Pierre Poulin, and Alain Fournier. "A survey of shadow algorithms". In: *IEEE Computer Graphics and Applications* 10.6 (1990), pp. 13–32.

[Xu+11] Li Xu et al. "Image smoothing via L 0 gradient minimization". In: *ACM Transactions on Graphics (TOG)*. Vol. 30. 6. ACM. 2011, p. 174.

[Zia07] Feras M Ziadat. "Effect of contour intervals and grid cell size on the accuracy of DEMs and slope derivatives". In: *Transactions in GIS* 11.1 (2007), pp. 67–81.