

# Visualization of Meteorological Data

I. Hakštok, Ž. Mihajlović

University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia  
ivan.hakstok@fer.hr, zeljkam@zemris.fer.hr

**Abstract** - This paper is focused on the methods used to visualize meteorological factors based on the given data. Properties of the factors for which the data can be gathered were described and the methods of visualizing those properties have been examined. The program solution is centered around interaction with the weather data. The data itself is based on real measurements. It is possible to select an exact point in time for which the data exists, and that data is then visualized on a 3D model of a building, which allows the user to get an idea of the weather state. The model changes lighting, shadows, clouds, opacity and the rotation of a wind vane, based on the data. The data is also represented by graphs, which allow the user to track data changes over a period of time. Finally, since the solution uses large amount of data, a performance analysis was done to point out advantages and disadvantages of the visualization methods used.

## I. INTRODUCTION

Meteorology is the scientific study of the atmosphere, and as such it is one of the most needed areas of science for everyday lives. It encompasses a vast array of factors which influence the behavior and appearance of the atmosphere.

The need for visualizing meteorological data comes from the fact that it's not easy for people to picture certain aspects of that data using numbers alone. However, those numbers are linked to the state of the atmosphere and visualizing them can give people a better and more clear understanding on the way they work and influence each other. This can be used in order to help plan construction of buildings or placing solar panels, where the effects of meteorological factors on surfaces need to be observed for longer time spans.

An important thing to notice in the visualization of the given data is that not all meteorological factors can be visualized directly, but it's possible to visualize the way they affect the surroundings, such as temperature causing heat blurs and mirages [1]. This paper describes some of the already existing ways to visualize given meteorological data using 2D and 3D graphics, along with graphs and charts.

## II. METEOROLOGICAL DATA

Some of the most important factors in meteorology are the temperature, humidity, atmospheric pressure, speed and the direction of the wind, and also the amount of rainfall. In order to visualize those factors, it's important to know how they interact and what effect they have on their surroundings.

It's possible to divide the meteorological factors into those that can be visualized directly and those that can't. Something like temperature of the air, which is the measure of the amount of heat energy present in the air, is not a visible phenomenon, but the effects of the temperature on the surroundings are noticeable: heat haze, ice and mirages are just some of the manifestations of the current air temperature.

Wind speed and direction are also not visible phenomena because humans can't see the movement of molecules in the air, but their interaction with the objects such as trees, leaves, flags and windmills can be observed and measured. Winds are a direct consequence of differing amounts of air pressure in two different places.

Some of the visible factors are cloud coverage, amount of rainfall and insolation. Those can be visualized directly, without the need for visualizing intermediate objects or phenomena. However, they are influenced by other factors which must be taken into consideration to some degree.

### A. Air Temperature

Air temperature is the most fundamental meteorological factor. It describes the amount of heat energy of air in a certain location. In terms of visualization, temperature can be visualized using graphs and heat maps, but in a 3-dimensional environment model, direct visualization isn't possible. However, air temperature does bring forth some interesting phenomena whose direct visualization in such a model is possible.

One of those phenomena is the appearance of so called mirages [1]. Mirages are a result of two layers of air with different density, which is a direct consequence of one layer being at a higher temperature than the other one. When air temperatures are high, heating of surfaces also occurs, and the air mass right next to the surface gets heated even further whereas the air further from the surface does not. The higher the air temperature is, the less dense the said air is, according to the ideal gas law [2]. The law states that

$$pV = nRT. \quad (1)$$

Assuming the volume, indicated by  $V$  in (1), is constant, the pressure  $p$  is inversely proportional to the temperature  $T$ .

Since the aforementioned two layers are of different density, light reflection occurs at the borderline of the two

layers when observed from a certain angle [3]. This results in a mirror effect for the viewer.

Another phenomenon affected by the temperature is the rainfall. Rainfall can vary between rain, snow and hail, and the type is largely dependant on the temperature. More of this will be discussed in the section focusing on visualizing rainfall.

### B. Wind

Air molecules by themselves aren't visible to the human eye, but the effect of their movement on the surroundings is easily noticeable and possible to visualize. Wind speed is measured using anemometers. Therefore, the anemometer can be used as a way of visualizing the strength and speed of the wind. Because there are various different types of anemometers, it's necessary for visualization to select a type that's easy to model and visualize.

Since the wind direction should also be taken into account when visualizing wind data, the most suitable type would be the windmill-shaped anemometer [4]. Such anemometer is a combination of a wind vane and an anemometer. Wind speed affects the rotation of the windmill part, and wind direction affects the direction of the wind vane. Such rotations (around an arbitrary axis parallel to the xy plane for the windmill, and an arbitrary axis parallel to the z-axis for the wind vane) are easy to model. It's safe to assume that the speed of the windmill wing tip is equal to the speed of the wind multiplied by a certain coefficient. The angular velocity of the windmill would then be

$$\omega = \alpha v/r, \quad (2)$$

where  $v$  is the speed of the wind, and  $r$  the length of a wing. The coefficient can be any number, so one can be chosen depending on the other parameters and what is best suited for the model itself.

If the vector of the wind velocity is assumed to be parallel to the surface, i.e. the xy plane, the axis of the windmill rotation is then a line whose vector of direction is equal to the wind velocity vector. Following the assumption about the vector being parallel to the xy plane, wind direction can be represented by the angle between the velocity vector and an arbitrary starting vector. The starting vector is usually the direction of North, placed at 0 degrees. East is placed at 90 degrees, South at 180 degrees and West at 270 degrees.



Figure 1. Example anemometer model

### C. Cloud Coverage

Speaking in the terms of meteorology on Earth, clouds are visible masses of water droplets suspended in the atmosphere [5]. Appearance of clouds happens almost daily, and the amount of cloud coverage is measurable. Scales for measuring cloud coverage typically range from 0 to 8 or from 0 to 10.

There exist various ways of modeling and visualizing clouds. Clouds can be represented either by solid models, by particle systems [9] or textures. In the terms of meteorology, since measurements of cloud coverage are taken from stations usually located on the ground, one can assume that the need for visualizing clouds will also happen in a model where the viewer is close to the ground, and therefore the most efficient solution is to use different textures for different amounts of cloud coverage.

Textures can be dynamically generated based on actual data or generated as an interpolation between minimum and maximum cloud coverage [7]. An example of texture interpolation using a modeling program called Blender3D can be seen in Fig. 2. Another option is to have pre-rendered textures for each degree of cloud coverage. This reduces the amount of calculation that needs to be done during the rendering of a model, but it's lacking in flexibility. Considering measurement scales for cloud coverage are relatively short and quantized, limitations of pre-rendered textures don't influence the look of the model as much as they would for a more dynamic model.

### D. Insolation

Insolation is the amount of solar radiation received by the surface. It is influenced by cloud coverage to a lesser extent. Insolation amounts vary greatly throughout a year because they are not solely dependent on the position of the sun.

Position of the sun is important in a model involving meteorological data, but it can't be calculated by measuring insolation. It's therefore necessary to calculate the position based on the current date. A possible formula for calculating the position is described in [8]. The formula yields two coordinates in a horizontal coordinate system which has the viewer in the center, the solar azimuth  $\gamma$ , which represents the angle between north and the projection of the solar vector on the surface, and the zenith distance  $\theta_z$ , which represents the angle between the zenith and the solar vector. Using these two coordinates, the exact position can be visualized. However, the distance of the sun from the viewer must be scaled along

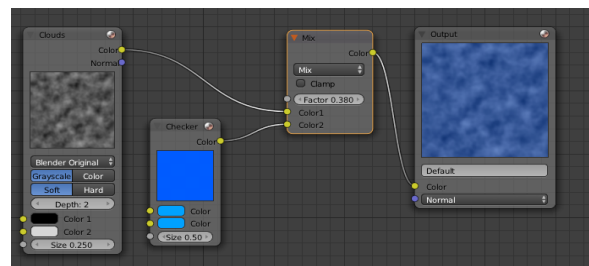


Figure 2. Texture interpolation between two textures in Blender3D

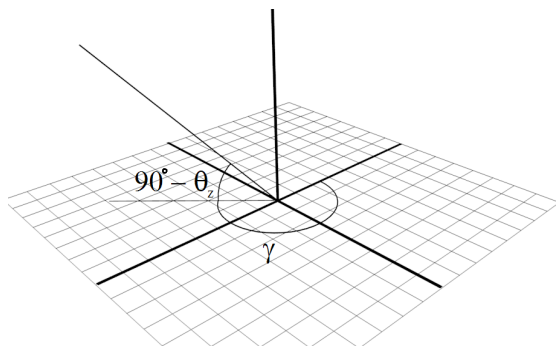


Figure 3. Solar coordinates in a horizontal coordinate system

with the illumination intensity in order to avoid using high distance values inside the model.

#### E. Rainfall

Rainfall can be visualized using a particle system. Measured amounts of rainfall can be related to the density of the particle system, while the vertical velocity of the particles can be assumed to be constant relative to the viewer and equal to the terminal velocity [10].

Type of rainfall varies between rain, snow and hail. Those types have different characteristics and causes, but it is safe to say that their terminal velocity is similar. The most prominent cause for the different types is the air temperature. Since the visualization is done based on real data, it is hard to deduce the type of rainfall from the data. It can be assumed that snow occurs at temperatures below zero whereas rain occurs at temperatures above zero. Hail is unpredictable and mostly short-term, making it redundant in terms of visualizing data collected on an hourly basis.

Another important factor in the visualization of rainfall is the influence of the wind. Strong winds make the rain and snow fall diagonally or sideways, increasing their horizontal velocities.

Particle systems make it easy to change the parameters for visualizing rainfall, such as particle velocity, vector of movement and the type and color of particles. If the clouds are represented with solid models, such models can serve as particle emitters. However, since clouds are often represented using textures, particle systems with invisible emitters are necessary.

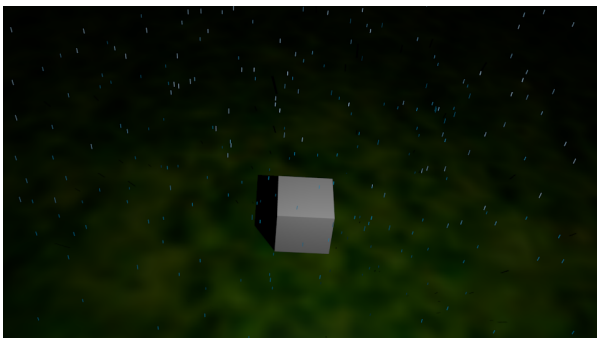


Figure 4. Rain as a particle system

#### F. Visibility

Visibility is related to the measurement of the distance in which an observer can make a distinction between objects or light sources. It is usually measured in kilometers and defined as the largest distance in which a black object can be seen in a light background [11].

Low visibility is usually caused by fog. Since an observer can move freely through the fog, it needs to be visualized in a way which distributes the fog relative to the observer. It can be visualized using particle systems or textures.

One downside of a particle system for fog visualization is the high amount of particles needed at low visibility. However, in modern computers, particle system fog visualization doesn't present a big problem and it results in more realistic scenes when combined with volumetric lighting [12].

Another way of visualizing fog is by using layers of semi-transparent textures. This method is fast and simple, but it cannot visualize advanced phenomena such as light scattering inside the fog.

### III. 3D MODEL WITH SIMULATED METEOROLOGICAL EFFECTS

In order to demonstrate some of the more simple methods for visualizing meteorological data, a program solution has been made as an example. The program solution supports both rendering a 3D environmental model based on meteorological data at a certain point in time and displaying data for large periods using graphs.

#### A. Input Data

Available input data consists of real measurements taken at hourly intervals during a larger period of time, and it's contained inside a text file. Text file header contains information about the meteorological station that measured the data: name, geographic coordinates (latitude, longitude) and height above sea level. Each line contains measured data, measured at the interval of one hour. Measured data consists of year, month, day, hour, average wind speed, wind direction, maximum wind speed, air temperature, relative humidity, air pressure, insolation, rainfall quantity, rainfall length, cloud coverage and visibility. Data can easily be accessed for a certain date and time.

#### B. Graphs

Graphs provide an insight into data, and allow for tracking of changes over a certain period of time. Range of the data selected for the graphs can be arbitrary, which allows for greater flexibility when observing data. An example graph is shown in Fig. 5. A viable option would be selecting the sampling frequency for the horizontal data range, but since performance is not an issue, such option is not considered.

All graphs consist of the date being displayed on the x axis and the value for the measured data being displayed on the y axis. One exception is the wind direction graph,

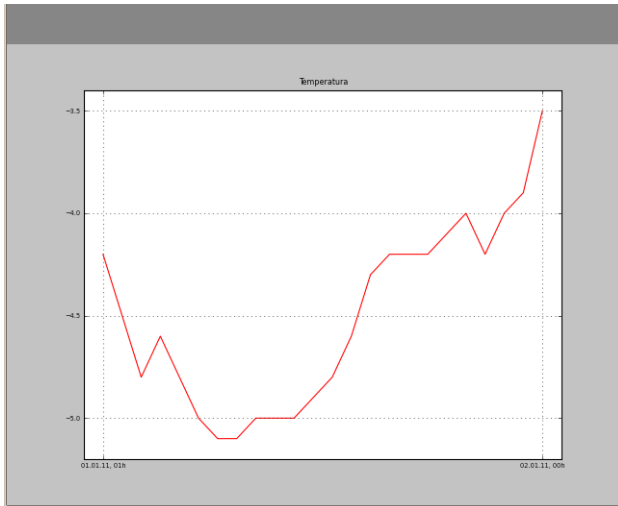


Figure 5. Air temperature graph

which instead displays the frequency of appearance of a wind direction in a polar graph.

### C. 3D Graphic Model

Modeling a scene in order to visualize data consists of modeling the scenery and modeling the visual representations of data.

The scenery can be modeled after the real world, e.g. a building is defined. Because the scenery is mostly static, size of the scenery does not play a large part in the performance of the model, but again that depends on the complexity of the methods used for visualization.

For this example, the tools used were Blender3D [6], for the modeling of scenery and additional objects, and Ogre3D [13], for displaying a rendered scene inside a window. Ogre3D uses scene and resource Managers to organize assets necessary for rendering a scene. Resource manager's concerns are importing textures, materials and scene configuration, while the scene manager organizes models, cameras, particle systems and lighting.

Inside the scene manager, it is necessary to initialize a frame listener. Frame listener tracks changes in data and is usually called once per frame, but the calling frequency can be changed. For this example, only change that happens each frame is the rotation of the anemometer. Other changes happen whenever the read input has changed.

In order to visualize the sky and clouds, it is necessary to add a sky dome, as pictured in Fig. 6. A sky dome is a hemisphere placed over the scene. Textures can be placed on the dome which results in the visualized scene to be surrounded with the sky texture.

Since there's a need for detailed tracking of the wind direction and speed, anemometer changes can be followed. That makes necessity for one camera to be focused on displaying only the anemometer. Simultaneous display of multiple cameras can be achieved using multiple viewports, and in this case one viewport is nested inside the other, as can be seen in Fig. 7.

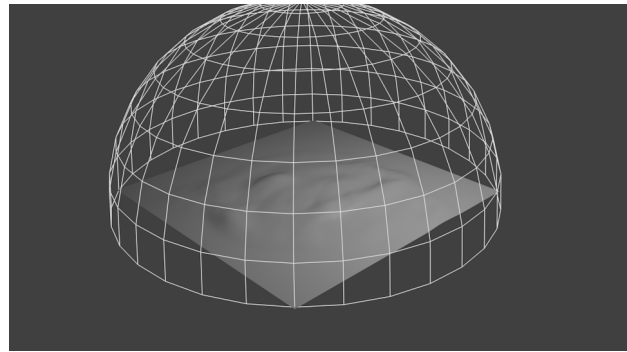


Figure 6. Sky dome wireframe over an arbitrary terrain

Another factor that plays a big role in the rendering of the scene is lighting. Position of the sun can be calculated, and therefore a directional light source can be placed inside the sky dome at the calculated coordinates. However, in small models, light source may continue providing ambient lighting for the model even after sunset. To counter this, ambient lighting must be set to darker values after sunset. Atmospheric refraction can also be simulated, which results in the sky having an orange hue near the time of sunset [14].

Ambient lighting is also dependant on the cloud coverage. The dependancy can be assumed as linear, even though it depends on the distribution of the clouds rather than the exact percentage of the sky covered.

## IV. CONCLUSION

Simpler visualization methods are far from realistic looking in terms of visualizing meteorological data, but provide an insight into the state of the atmosphere at which the data were collected.

There are lots of ways in which various meteorological factors are related to each other, and therefore it is hard to make an accurate model based on small amounts of data. Still, based on the given data, some relations can be noticed and visualized.

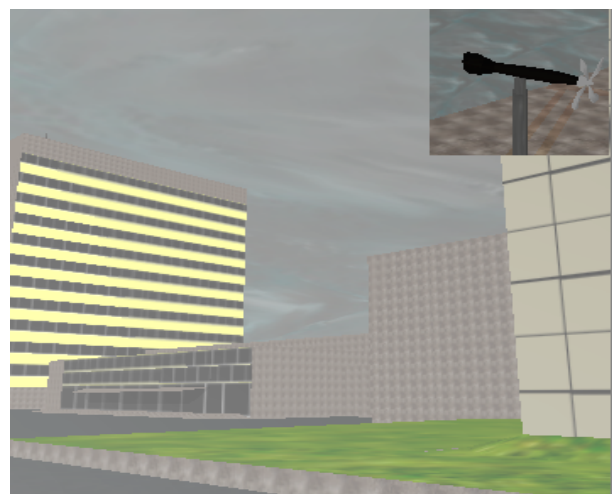


Figure 7. Dual-viewport display

#### ACKNOWLEDGMENTS

This work has been financially supported by the Croatian Science Foundation under grant No. I-4463-2011 (MICROGRID) and by the European Community Seventh Framework Programme under grant No. 285939 (ACROSS). This support is gratefully acknowledged.

#### REFERENCES

- [1] A. T. Young, An Introduction to Mirages, <http://mintaka.sdsu.edu/GF/mirages/mirintro.html>, 2013.
- [2] R. Hochstim, Gases, <http://www.premed411.com/pdfs/CHM456.pdf>, 1996.
- [3] Reflection and Refraction of Light, <http://www.physics.uc.edu/~sitko/CollegePhysicsIII/22-ReflectRefract/ReflectRefract.pdf>, 2014.
- [4] Anemometer, <http://www.saylor.org/site/wp-content/uploads/2011/04/Anemometer.pdf>, 2013. Weather Terms Glossary, <http://www.erh.noaa.gov/box/glossary.htm>, 2014.
- [5] Weather Terms Glossary, <http://www.erh.noaa.gov/box/glossary.htm>, 2014.
- [6] Blender 3D, <http://www.blender.org/>, 2014.
- [7] R. Ruiters, R. Schnabel, R. Klein, "Patch-based Texture Interpolation", Eurographics Symposium on Rendering (Proc. of EGSR), vol. 29, 2010, pp. 1421-1429
- [8] M. Blanco-Muriel, D. C. Alarcón-Padilla, T. López-Moratalla, M. Lara-Coira, "Computing the Solar Vector", Solar Energy Vol. 70, No. 5, pp. 431-441, 2001.
- [9] W. T. Reeves, "Particle Systems— a Technique for Modeling a Class of Fuzzy Objects", ACM Transactions on Graphics (TOG), Vol. 2, No. 2, pp. 91-108, April 1983.
- [10] G. B. Foote, P. S. du Toit, "Terminal Velocity of Raindrops Aloft", April 1969.
- [11] W. C. Malm, Introduction to Visibility, Cooperative Institute for Research in the Atmosphere (CIRA, NPS Visibility Program), Colorado State University, Fort Collins, CO, 2000. <http://www.epa.gov/air/visibility/pdfs/introvis.pdf>, 2013.
- [12] B. Tóth, T. Umenhoffer, "Real-time Volumetric Lighting in Participating Media", EUROGRAPHICS 2009.
- [13] Ogre API Reference, <http://www.ogre3d.org/docs/api/html/>, 2014.
- [14] M. E. Thomas, R. I. Joseph, Astronomical Refraction, <http://techdigest.jhuapl.edu/TD/td1703/thomas.pdf>, 1996.