Legal Statement

1. The copyright of this document is covered by SuperMap Software Co., Ltd. in accordance with the Copyright Law of the People’s Republic of China and the Universal Copyright Convention. If, without the written permission of the company, any part of the document shall not in any way or any reason be used, copied, modified, transmitted, or bundled with other products to be used, sold, tort reserved.

2. “超图”, “SuperMap”, and are the registered trademarks of SuperMap Software Co., Ltd., protected by the Copyright Law of the People’s Republic of China. If, without the written permission of the company, the trademarks shall not in any way or any reason be used, copied, modified, transmitted, or bundled with other products to be used, sold, tort reserved.

3. This document represents no responsibilities of any supplier or agent. Without statement, SuperMap Software Co., Ltd. has right to do modifications to this document.

4. The copyright of trademarks mentioned in this document belongs to the corresponding companies. Without the written permission of these companies, the trademarks shall not in any way or any reason be used, copied, modified, or transmitted.

5. The concerned software products and the updated products hereinafter in this document are developed and sold by SuperMap Software Co., Ltd.

Hereby declare

SuperMap Software Co., Ltd.:  
Add: 7/F Tower B, Technology Fortune Center, No. 8 Xueqing Road,  
Haidian District, Beijing, 100192, P. R. China  
Tel: +86-10-82736655-4170  
Fax: +86-10-82734630
HomePage: www.supermap.com

Sales: request@supermap.com

Tech Support: globalsupport@supermap.com

SuperMap Software welcomes all advices and suggestions from you.
Content

1 Summary .......................................................................................................................... 1
  1.1 Why it is necessary to use realspace? ........................................................................... 1
  1.2 Features of Realspace ................................................................................................ 3

2 Two and Three Dimensional Data Integration ............................................................... 1
  2.1 Vector Dataset .......................................................................................................... 2
  2.2 KML Data ................................................................................................................ 6
  2.3 Geometric Objects .................................................................................................... 8
    2.3.1 Three Dimensional Geometric Objects .............................................................. 9
    2.3.2 Regular Three Dimensional Geometric Objects ............................................... 11
  2.4 Third Party's 3D Model ............................................................................................. 12
  2.5 Massive Image Data .................................................................................................. 13
  2.6 Terrain Data ............................................................................................................. 15

3 Integration of Two and Three Dimensional Displaying .................................................... 1
  3.1 Layers ........................................................................................................................ 1
    3.1.1 The Ordinary Layer ............................................................................................ 1
    3.1.2 Three Dimensional Tracking Layer ................................................................... 2
    3.1.3 Screen Layer ..................................................................................................... 3
    3.1.4 Terrain Layer .................................................................................................... 5
  3.2 Maps .......................................................................................................................... 5
  3.3 Thematic Maps ......................................................................................................... 6

4 Integration of Two and Three Dimensional Analysis ....................................................... 1
  4.1 Surface Measurement ............................................................................................... 1
  4.2 Network Analysis ....................................................................................................... 2
  4.3 Simple Query ............................................................................................................ 3
  4.4 SQL Query ................................................................................................................ 4
  4.5 Spatial Query ............................................................................................................. 4
5 Basic Concepts in Scene

5.1 General Introductions of Three Dimensional Window and Scene

5.2 Camera

5.3 Altitude Mode

5.4 Texture Mapping

6 Three Dimensional Operations
Summary

GIS enables people to query, analyze, process and display a variety of information with a real-world view, working out scientific strategies. After more than a decade development, the two dimensional GIS technique has been widely applied in several dozen industries, promoting business management and work efficient dramatically. In the 2D GIS, spatial positions are represented by (x, y); while in the 3D GIS, that is (x, y, z). Hence, the latter provides a possibility of displaying objects as they are and analyzing methods and functionality in 3D environments, which are also the desires of many people. Furthermore, in many situations, for instance, the combat command, terrain simulation, digital cities, real estate exhibition, environment protection, meteorology, atmosphere pollution, etc, it is only the 3D GIS that can offer optimum performance.

To date, the 3D GIS technology has been not only widely applied in practice, but also has become an industry which contributes to a significant revolution of GIS technology. However, the data model of 2D GIS provides more simple, abstract, and comprehensive approaches. As a result, either of them can be independent to meet various demands of today's customers. Consequently, the correct development direction of GIS should combine both of them together, which is necessary to introduce a new concept - the Realspace GIS.

The Realspace GIS is used in contrast to the paperspace that is the projected paper space. As a result, the paperspace GIS is the GIS software that is on the basis of the projected paper space. The "realspace" is a combination of 3D geographic space, and a 2D map space which is on the basis of geographic sphere or ellipsoid. Similarly, the Realspace GIS is the GIS software that is on the basis of the combined environment.

1.1 Why it is necessary to use realspace?

3D GIS has many advantages that the 2D GIS does not possess:

(1) Real-world displaying
In 3D GIS environments, without projection work in 2D GIS, the true nature of the earth can be displayed as it is, for instance, the rise and fall of the terrain, the objects in the space or underground, as well as the light, fog, water stream phenomena. 3D GIS enables GIS products to be widely applied for the public.

(2) Spatial analyzing

3D GIS has capabilities in terrain analysis, sunlight analysis, spatial spreading analysis, visible analysis, submerge analysis, all of which provide the optimum approach to interpret information and spatial analysis, providing support to make decisions in all industries.

The necessary of 2D GIS:

With the development of CPU technique and high performance visualization algorithm, the visibility speed of large scale scene is promoted dramatically. Furthermore, the progress of aerial and close range photogrammetry has laid solid data foundation for the development of 3D GIS to be applied in practice. However, 2D GIS, with a long history since 1960s, is still needed by people, because of its traditional functionality, such as various mapping functions; good performance to provide many query approaches and analysis ways; data processing ability and simple data models, and applications in many businesses. Consequently, the current 3D GIS, which is mainly used to view, is not possible to replace 2D GIS totally at once.

Moreover, the cost of obtaining 3D spatial data is higher, especially in building large scale scene. Secondly, 3D data model is more complicate, contributing to low efficiency of spatial query and analysis functionality. At last, because of the limitation of network transmission and massive data management, 3D GIS cannot provide real-time information yet.

Realspace: Integration of 2D and 3D

With the advantages of 2D and 3D, it is desired to have both of them together in a system. However, this kind of product usually has independent two systems in nature, a 3D system and a mainstream large scale 2D GIS platform, even with the displaying function combining both 2D and 3D. Precisely speaking, it is obvious on the aspects of data, displaying approach, and analysis functionality, which generate high cost and great difficulty in GIS applications. Hence, we should start from the GIS platform bottom technology to solve this problem.
1.2 Features of Realspace

The technology of Realspace offered by SuperMap does not need any projection, but load data with latitude and longitude to the 3D spherical surface directly, which is a real 3D geographic space displaying. Realspace is built on the basis of SuperMap, with the cutting edge of computer software and hardware techniques, as well as visualization technique, providing more powerful functionality, higher performance and vividness.

The Realspace GIS is not just a three dimensional environment, but an integration of two dimension and three dimension:

(a) The data model and data structure of two and three dimension keeps in integration. All two dimensional data requires no transmission processing, offering high performance of visibility in scene. It not only requires the 3D GIS to be compatible with the two dimensional data structure, but also the latter to adjust to realize the visualization of massive two dimensional data in the three dimensional scene.

(b) All analysis methods and processing functionality in 2D GIS can be applied in the three dimensional scene directly.
Two and Three Dimensional Data Integration

The data model and data structure of two dimensional and three dimensional keeps in integration. The 3D GIS data has two dimensional data structure, and the two dimensional data also adjust to viewed in three dimensional scene.

The 3D GIS of SuperMap Objects Java is on the basis of OGDC standard (Open Geo-DataBase Connectivity), which makes the accessing of data from different sources possible, displaying data in different formats in the same scene. For instance, directly opening the data format supported by SuperMap SDX+; supporting standard format and model data, such as WMS, WFS, WMS, KML\KMZ.

In this chapter, the three dimensional data supported by SuperMap Objects Java is going to be introduced. Please look at Diagram 1.

![Diagram 1 Three dimensional data model supported by SuperMap Objects Java](image)

SuperMap Objects Java is able to display two dimensional data in three-dimensional way, including adding two dimensional geometric objects and various two dimensional vector layers into three
Two and Three Dimensional Data Integration

In the three dimensional scene. Meanwhile, it supports processing two dimensional data into three dimensional data, for instance, stretching two dimensional objects into three dimensional regions and three dimensional bodies, as well as the rendering of texture. For the three dimensional data, it can import three dimensional geometric objects as well as the third party three dimensional model into three dimensional scene. The 3D GIS of SuperMap Objects Java also supports adding KML files into three dimensional scene as three dimensional layers.

Additionally, the 3D GIS of SuperMap Objects Java provides preprocessing and loading of massive image data and terrain data, realizing high management efficiency of massive data.

The three dimensional data in the scene is organized as three dimensional layers, in other words, adding three dimensional data is actually adding data to the corresponding layers. The layers are defined as 'Layer3D' type layer, screen layer (ScreenLayer3D), terrain layer (TerrainLayer), and 3D tracking layer (TrackingLayer3D). In the three dimensional scene, there is only one screen layer, one 3D tracking layer, but it allowed to have multiple layers in "layer3D" type which is managed by 3D layer set (class Layer3Ds) and multiple terrain layers that is managed by terrain layer set (class TerrainLayers).

In the following sections, the three dimensional data will be introduced in detail by using various data and layers supported by 3D GIS of SuperMap Objects Java. Furthermore, how to add data into the correct layers will also be illustrated.

Remarks: the three dimensional layers in 'Layer3D' type are classified as 3D dataset layer, 3D map layer, KML layer, image file layer and model cache layer.

2.1 Vector Dataset

In the three dimensional scene of 3D GIS in the SuperMap Objects Java, the SuperMap 2D vector data can be directly opened to browse. The vector data that could be used are: point dataset, line dataset, region dataset, route dataset, network dataset, and geometric objects of points, lines, and regions in the compound dataset.

3D GIS in SuperMap Objects Java can add the existing two dimensional vector dataset into the scene to be a layer, which is the conversion procedure from 2D to 3D. Every geometric object of the
transformed 3D layer is represented by coordinates X, Y, Z, i.e. the two dimensional points, lines, regions become three dimensional ones. Meanwhile, you may stretch the geometric object in 2D dataset, and the height of stretching could be customized. The elevation information of the transformed geometric object in 3D layer could be specified. Users are also able to set the displaying style and render the 3D objects. In nature, the conversion is on the basis of 2D coordinate information, putting them on the sphere in the scene.

The elevation value should be set for the 2D vector dataset before loading into the scene. There are three setting ways: first, follow the sphere, which is displaying the dataset by clinging to the ground; second, set the relative height value D, which is the height to the ground; third, set the absolute height, which is the distance from the geocenter. By default, the dataset will be displayed by following the sphere. Please refer to the chapter of elevation pattern.

Remarks: currently, the 2D vector dataset can only be displayed in the scene, but cannot be saved in it.

During the conversion procedure, the specification of the elevation information, stretch height, and display style should be done through a class named Layer3DSettingVector. The detail information of the Layer3DSettingVector class is showed in Table 1:

<table>
<thead>
<tr>
<th>METHODS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>get/setBottomAltitudeField</td>
<td>Obtains or sets of the bottom elevation field name. When displaying the objects of the vector dataset in a 3D window, both the original coordinate information and elevation value in attribute field can be used to illustrate the objects on the correct position.</td>
</tr>
<tr>
<td>get/setExtendedHeightField</td>
<td>Obtains or sets the stretching height field. When vertically stretching the 2D objects, the stretching height value will be used to decide the stretched extent.</td>
</tr>
<tr>
<td>get/setSideTextureField</td>
<td>Obtains or sets the side texture field name, which stores the storage routes of texture images separated by semicolons. When the 2D</td>
</tr>
</tbody>
</table>

Remarks: currently, the 2D vector dataset can only be displayed in the scene, but cannot be saved in it.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>get/setStyle</td>
<td>Obtains or sets the style of the 3D layer transformed from vector dataset to merge into the 3D window.</td>
</tr>
<tr>
<td>get/setTilingUField</td>
<td>Obtains or sets the side texture's horizontal repetition times. When the 2D data is vertically stretched into 3D solid objects, they will apply the specified images to rendering texture. The system will stretch the pictures according to pictures' dimensions, the solid objects' dimensions, and the side texture mapping (vertically and horizontally) repetition times. The horizontal repetition times is the value of this attribute.</td>
</tr>
<tr>
<td>get/setTilingVField</td>
<td>Obtains or sets the side texture's vertical repetition times. When the 2D data is vertically stretched into 3D solid objects, they will apply the specified images to rendering texture. The system will stretch the pictures according to pictures' dimensions, the solid objects' dimensions, and the side texture mapping (vertically and horizontally) repetition times. The vertical repetition times is the value of this attribute.</td>
</tr>
<tr>
<td>get/setTopTextureField</td>
<td>Obtains or sets the top texture field name, which records the storage routes of the texture images separated by semicolons. When the 2D data is vertically stretched into 3D solid objects, the top of objects will apply the image specified by this attribute to render.</td>
</tr>
<tr>
<td>getType</td>
<td>Obtains the type of 3D layer extension setting, in other words, to obtain the correct three dimensional layer extension setting class for the object.</td>
</tr>
</tbody>
</table>

Steps of adding two dimensional vector dataset into the three dimensional scene:

Obtain the scene that is going to contain the 2D dataset;
Set vector dataset's 3D layer extension setting (class Layer3DSettingVector);

Call the method, put the 2D vector dataset into the scene, as a dataset type 3D layer;

Syntax:

Layer3DDataset Scene.getLayers().add (Dataset dataset, Layer3DSetting layer3DSetting, Boolean addToHead)

Parameters:

dataset is designated 2D vector data.

layer3DSetting is the 3D layer extension setting object of the designated vector dataset. The type of Layer3DSetting is Layer3DSetting. It is the base class of Layer3DSettingVector class.

addToHead decides if the new added 3D layer is put on the top of others.

Return Value:

3D dataset layer class objects.

Diagram 2 A line dataset is added into the scene (with stretching and mapping)
2.2 KML Data

KML (Keyhole Markup Language) is a file created on the bases of XML language and file format, using the structure that contains embedded elements and attributes on the basis of markup. All the markups are case sensitive. KML is used to describe and store geographic information, for instance, points, lines, images, polylines, and models, displaying them in the application system. For instance, the Google Earth, Google Map and so on. KML is the international standard supported by OGC (Open Geospatial Consortium). The file of KML has two extension names: *.kml and *.kmz (compressed set of one or multiple KML files that is in zip format).

The layer types supported by KML include geometric objects (points, lines, and regions), pictures and models. In the scene, the geographic information recorded by KML files is the KML 3D layer, which means the information recorded by KML file is added on the sphere according to the coordinate information.

Steps of adding KML file into the scene:

1. Obtain the scene that is going to contain KML files;
2. Obtain the desired KML files;

3. Call the method to add the KML files into the scene, as a KML 3D layer;

Syntax:

Layer3D Scene.getLayers().add(String dataName, Layer3DType layerType, Boolean addToHead)

Parameters:

dataName is the full directory of the data that is corresponding to the specified Model, KML, Map or Image File layer.

layerType is the type of layer. If the added data name does not match to the layer type, then the adding is breakdown.

addToHead specifies if putting the new added 3D layer to the topmost.

Return Value:

3D layer object (return the object of Layer3DKML 3D layer).
2.3 Geometric Objects

3D GIS in SuperMap Objects Java may add multiple geometric objects into scene. To date, it only can add 3D geometric objects into 3D layer in the scene, as 3D features class (Feature3D class). 3D features class in the 3D layer are managed by 3D features set (Feature3D class). Besides, the structure of Feature3Ds is a tree construction. Please refer to the relative interfaces in the Programmer Reference.

The 3D geometric objects added into KML 3D layer will be put on the scene sphere, according to their coordinate information.

Because of the limitation of 3D conversion achievement approaches developed by 3D GIS of SuperMap Objects Java, the ways to add different 3D geometric objects into the scene are not totally the same. Therefore, the 3D geometric objects are divided, according to the import method, into points, lines, regions, images, text, and regular three dimensional geometric objects. The following paragraphs will introduce how to cope with them one by one.
2.3.1 Three Dimensional Geometric Objects

The displaying style of three dimensional points (GeoPoint3D), lines (GeoLine3D), regions (GeoRegion3D) that are added to the 3D layer could be set. Please refer to Table 2 to know about the setting of 3D displaying style.

Table 2 Methods of Three Displaying Style Class (GeoStyle3D class)

<table>
<thead>
<tr>
<th>METHODS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>get/setAltitudeMode</td>
<td>Obtains or sets the altitude mode of the 3D geometric object. Please refer to the class AltitudeMode.</td>
</tr>
<tr>
<td>get/setBottomAltitude</td>
<td>Obtains or sets the bottom elevation value. When displaying the objects of the vector dataset in a 3D window, the original coordinate information can be used to illustrate the objects on the correct position. Meanwhile, the bottom elevation value, provided by this attribute, can be used to display the object to certain height.</td>
</tr>
<tr>
<td>get/setExtendedHeight</td>
<td>Obtains or sets the stretching height value. When vertically stretching the 2D geometric objects that are going to be displayed in the 3D window, the stretching height value, recorded by this attribute, will be used to decide the stretched extent.</td>
</tr>
<tr>
<td>get/setFillForeColor</td>
<td>Obtains or sets the fill foreground color of the region geometric objects.</td>
</tr>
<tr>
<td>get/setFillMode</td>
<td>Gets or sets the fill mode of 3D geometric solid objects, including the outline filling, region filling, outline and region filling. Please refer to FillMode3D class.</td>
</tr>
<tr>
<td>get/setLineColor</td>
<td>Gets or sets the color of 3D line geometric objects.</td>
</tr>
<tr>
<td>get/setLineWidth</td>
<td>Gets or sets the width of 3D line geometric objects.</td>
</tr>
<tr>
<td>get/setMarkerColor</td>
<td>Gets or sets the color of 3D point geometric objects.</td>
</tr>
<tr>
<td>get/setMarkerIconFile</td>
<td>Obtains or sets the full storage directory of 3D point geometric icons, which represent 3D point geometry object.</td>
</tr>
<tr>
<td>get/setMarkerIconScale</td>
<td>Obtains or sets the zoom ratio of point geometric object icons.</td>
</tr>
<tr>
<td>get/setMarkerSize</td>
<td>Obtains or sets the size of point geometric objects. The unit is 0.1mm.</td>
</tr>
</tbody>
</table>
get/setSideTextureFiles Obtains or sets the full storage directory of a series of images that are going to be used to render as sides. The sides of 3D solid objects will apply the image directed by this attribute to conduct texture render.

get/setTilingU Obtains or sets the side texture's horizontal repetition times. The sides of 3D solid objects will apply specified image to conduct texture render. The system will stretch the pictures according to pictures' dimensions, the solid objects' dimensions, and the side texture mapping (vertically and horizontally) repetition times. The horizontal repetition times is the value of this attribute. The type of this attribute is Double, i.e., the repetition times could be decimal.

get/setTilingV Obtains or sets the side texture's vertical repetition times. The sides of 3D solid objects will apply specified image to conduct texture render. The system will stretch the pictures according to pictures' dimensions, the solid objects' dimensions, and the side texture mapping (vertically and horizontally) repetition times. The vertical repetition times is the value of this attribute. The type of this attribute is Double, i.e., the repetition times could be decimal.

get/setTopTextureFile Obtains or sets the full storage directory of the texture images separated by semicolons. The top of 3D solid objects will apply the designated image to render.

3D geometric objects have to be added into the KML 3D layers through 3D landmark geometric object (GeoPlacemark). That is to say, first, the value of 3D geometric objects should be assigned to the attribute of 3D landmark geometric objects; second, add the landmark geometric objects into KML 3D layers, as 3D feature objects.

Steps to add 3D geometric objects into KML 3D layers:

1. Obtain one 3D layer in the scene that is going to receive 3D geometric objects;
2. Obtain the 3D points, lines or regions geometric objects that are going to be added. You may set their displaying style;
3. Obtain 3D landmark geometric objects, the attribute Geometry of 3D landmark geometric objects will get the value of 3D geometric objects;
(4) Obtain 3D feature element set objects that are going to contain 3D geometric objects. In other words, to decide which 3D feature element set (a tree organization) class in 3D layers should accept the 3D geometric objects;

(5) Call method add() of desired essential set class, add() is able to add the 3D landmark geometric objects which contains 3D geometric objects into 3D layers;

Syntax:

Feature3D Feature3Ds.add(Geometry3D geometry)

2.3.2 Regular Three Dimensional Geometric Objects

Regular 3D Geometric Objects offered by SuperMap Objects Java include pyramids, cones, truncated cones, hemispheroids, cake cones, 3D sectors, cuboids, and so on. Before adding them into 3D layers, we have to model them first.

SuperMap Objects Java offers interfaces to model these geometric objects, i.e. method Geometry.getGeoModel() which will obtain the geometric objects' 3D model, i.e. GeoModel class object.

Steps of adding regular 3D geometric objects into KML 3D layers:

(1) Obtain one KML 3D layer of scene that is going to receive 3D geometric objects;

(2) Obtain the regular 3D geometric objects, setting their displaying style;

(3) Model three dimensional regular geometric objects, i.e. obtains 3D model geometric objects of the regular three dimensional geometric objects;

(4) Obtain 3D essential element set objects that are going to contain 3D geometric objects. In other words, to decide which 3D essential element set (a tree organization) class in 3D layers should accept the 3D geometric objects;

(5) Call method add() of desired essential set class, add() is able to add the 3D landmark geometric objects which contains 3D geometric objects into 3D layers;

Syntax:
Two and Three Dimensional Data Integration

SuperMap Objects Java Technology Documents

Feature3D Feature3Ds.add(Geometry3D geometry)

Diagram 5 3D Regular Geometric Objects in Three Dimensional Scene

2.4 Third Party's 3D Model

SuperMap Objects Java offers 3D model geometric object class, the GeoModel class, which is able to import the third party data model, including *.3DS, *.SGM, and so on. The necessary interfaces are:

Boolean GeoModel. fromFile(String file)

Boolean GeoModel. fromFile(String file, Point3D position)

The GeoModel class object only can be added into 3D layers of the scene.

Steps to add 3D model geometric objects:

(1) Obtain one 3D model layer that is going to receive 3D models;

(2) Obtain the three dimensional model geometric objects that need to be added;

SuperMap Objects Java Technology Documents
(3) Obtain 3D feature element set objects that are going to contain 3D model geometric objects. In other words, to decide which 3D essential element set (a tree organization) class in 3D layers should accept the 3D model geometric objects;

(4) Call method add() of desired feature set class, add() is able to add the 3D model geometric objects into 3D layers.

Diagram 6 3D Regular Geometric Objects in Three Dimensional Scene

2.5 Massive Image Data

We have to preprocess the massive image data by creating a cache directory before putting them into the scene. It is just like cutting the massive image into layers and blocks based on global grid partitioning model, and then we got a cache directory and a descriptive SCI3D file (*.sci3d), which holds the directory’s information, such as layer number, file name, and sample dimensions and so on. 3D GIS in SuperMap Objects Java imports the massive image data by loading the SCI3D file, taking the massive image as a 3D layer SCI3D file in scene. Furthermore, SuperMap will automatically choose the most suitable number of layers and blocks to show the image, according to users’ displaying scale. This will improve the loading and browsing speed significantly.

3D GIS of SuperMap Objects Java can preprocess images in *.bmp, *.gif, *.jpg, *.png, *.tiff format, generating 3D GIS images that could be loaded to SuperMap Objects Java.
The coordinate information of the image will bring it to the right place on the virtual earth surface in scene.

Steps to load images by SCI3D file:

(1) Obtain the original image data, Image dataset;

(2) Generate image cache from Image dataset through CacheBuilderSCI3D class. Please refer to Programmer Reference for the detail of CacheBuilderSCI3D class.

(3) Obtain the scene that is going to contain image data;

(4) Set extension configuration of 3D image layer, i.e. class Layer3DSettingSCI object. Please refer to Table 3 about its attributes;

(5) Call the method Layer3Ds.add(), add image data into scene through SCI3D file.

Syntax:

Layer3D_Scene.Layers.add(String dataName,Layer3DType layerType, Boolean addToHead, String layerName)

Parameters:

file is the full directory route of the designated SCI3D file.

addToHead decides if the new added 3D layer is put on the top of others.

Return Value:

Return the new added 3D layer.

<table>
<thead>
<tr>
<th>METHODS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>is/setTransparent</td>
<td>Obtains or sets if using transparent color and the colors within the tolerance scope transparently.</td>
</tr>
<tr>
<td>get/setTransparentColor</td>
<td>Obtains or sets transparent color.</td>
</tr>
<tr>
<td>get/setTransparentColorTolerance</td>
<td>Obtains or sets the tolerance of the transparent color, with</td>
</tr>
</tbody>
</table>
2.6 Terrain Data

Terrain data displays the terrain wave of earth surface by elevation data. It is very useful in terrain analyses, such as viewshed analysis, dynamic analysis and simulation of flood undulation, perspective analysis and so on. This 3D GIS version in SuperMap Objects Java offers loading and browsing terrain data functions. The analyzing functions are not provided in this version.

In order to get terrain data supported by SuperMap Objects Java, the data should be preprocessed. SuperMap can preprocess the elevation data in DEM and Grid format, cutting the elevation data into layers and blocks based on global grid partitioning model. Finally, we got a cache directory and a descriptive SCT file (*.sct), which holds the directory's information, such as the number of cache layer, geographic scope, file type and so on. 3D GIS in SuperMap Objects Java imports the terrain data by loading the SCT file, taking the terrain as a 3D layer SCI file in scene Furthermore,
SuperMap will automatically choose the most suitable number of layers and blocks to show the image, according to users' displaying scale. This will improve the loading and browsing massive terrain data speed significantly.

According to coordinate reference information, terrain data will be added on the 3D sphere in the scene, with realistic terrain fluctuation.

The terrain layer in the scene is managed by terrain layer set, which can add, delete, and adjust the sequence of terrain layers.

Steps to load terrain data through SCI file:

1. Obtain the cache directory of preprocessed terrain data and its SCT file;
2. Obtain the scene that is going to contain the terrain data, and terrain layer set of the scene;
3. Call the method: load terrain data into scene through SCT file, i.e. adding it into the terrain layer set of scene, as one terrain layer;

Syntax:

TerrainLayer Scene.getTerrainLayers().add(String file, Boolean addToHead)

Parameters:

file is the full directory route of the designated SCT file.
addToHead decides if the new added 3D layer is put on the top of others.

Return Value:

Return the new added terrain layer.

Additionally, 3D GIS of SuperMap Objects Java is able to display terrain exaggeration, enlarging the fluctuation of terrain. The interface to achieve terrain exaggeration is Scene.TerrainExaggeration attribute. Please look at Diagram 9 for the effect of terrain exaggeration, with 3 as the coefficient of exaggeration.
Diagram 8 Terrain data covered by image data

Diagram 9 The exaggerated terrain data (coefficient of exaggeration is 3)
Integration of Two and Three Dimensional Displaying

In the scene, the data is loaded to the sphere as a layer as the two dimensional map does. For instance, the image data, terrain data, vector data are loaded into different layers. The layers are divided into three types: the ordinary layer, the tracking layer, as well as the screen layer and terrain layer.

3.1 Layers

3.1.1 The Ordinary Layer

The ordinary layer is used to load data to display. In scene, it allowed to store multiple ordinary layers with customize style. Whenever the scene is opened, the layer will be automatically loaded in the style set last time.

The position of the object will rotate with the sphere in the scene.

The ordinary layer consists of five types: dataset, map, KML, image, and model cache. The following is the introductions of them:

The 3D layer in dataset type: the 3D dataset that is generated from the 2D dataset, including vector dataset, raster dataset, and image dataset.

The 3D layer in map type: the 3D layer generated by dynamically projecting the 2D map into the scene.

The 3D layer in image type: the 3D layer that is generated from massive image data, the image
cache file (*.SCI3D or *.SIT) to the scene.

The 3D layer in KML type: the 3D layer generated from files in *.KML or *.KMZ format.

The 3D layer in model cache: the 3D layer generated from 3D model cache file (*.SCM).

Please refer to the "Two and Three Dimensional Data Integration" for how to add 3D objects to 3D ordinary layer.

### 3.1.2 Three Dimensional Tracking Layer

Three dimensional tracking layer is a temporary layer covering around the virtual earth surface in scene, containing the temporary 3D geometric object that is added in scene. When you close the scene, the content stored by 3D tracking layer will be cleared.

There is only one 3D tracking layer in one scene. Users are allowed to add or delete any kinds of 3D geometric objects.

Steps to add 3D geometric objects into 3D tracking layer:

1. Obtain the scene that is going to display objects;
2. Set the required 3D geometric objects;
3. Obtain the three dimensional tracking layer of the target scene. Add the set 3D geometric object to the 3D tracking layer;

Syntax:

```java
Int32 Scene.getTrackingLayer3D().add(Geometry geometry, String tag)
```

Parameters:

- `geometry` is the 3D geometric object that is going to be added.
- `tag` is the tag of the specified 3D geometric object that is going to be added.

Return Value:

Return the index value of the new added 3D geometric objects in its 3D tracking layer.
3.1.3 Screen Layer

3D GIS in SuperMap Objects Java provides a special layer, named screen layer. The positions of objects on screen layer do not depend on their coordinate information to be put on the virtual earth but on the 3D window. Therefore, these objects will not rotate; neither inclines with the virtual earth, but will be able to zoom in and out with the 3D window. That is to say, they are relatively static with 3D window. Hence, the information of logo, descriptive words that have to be static on the 3D window could be put on this layer.

Each scene is allowed to have only one screen layer. The operations of adding any 3D geometric objects, setting their displaying position, size, or deleting the unwanted ones are available in screen layer.

The following is an instance of adding the SuperMap logo picture, to illustrate the steps of adding objects on screen layer.

Steps to add objects on screen layer:

1. Obtain the scene that is going to display objects;

2. Create a new GeoPicture3D class object, and transfer the route value of SuperMap Logo picture to attribute ImageFile of GeoPicture3D class object. The size or position information could also be changed according to requirements;

3. Obtain screen layer of the target scene; add the geometric objects (GeoPicture3D class object) on the screen layer;

Syntax:

Int32 Scene.get ScreenLayer3D().add(Geometry geometry, String tag)

Parameters:

gallery is the required 3D geometric object that is going to be added.

tag is the label of the designated 3D geometric object.

Return Value:
Return index value of the new added 3D geometric objects in its screen layer.

```java
int32 Scene.getScreenLayer3D().add(Geometry geometry, String tag, Boolean isLocationFixed, Boolean isSizeFixed)
```

Parameters:

Geometry the required 3D geometric objects that are going to be added.

Tag is the label of the designated 3D geometric object.

isLocationFixed desides if the position of 3D geometric object is fixed. The "true" indicates it is fixed, "false" means not.

isSizeFixed desides if the size of required 3D geometric object is fixed, represented by "true", or not by "false".

Return Value:

Return the index value of the new added 3D geometric objects in its screen layers.
3.1.4 Terrain Layer

3D GIS of SuperMap Objects Java also provides terrain layer. The terrain data in the scene is managed as terrain layer, stored as terrain cache file data, *.SCT. The terrain layers are managed through terrain layer set, which can add, delete, and adjust terrain layer's sequence.

3.2 Maps

SuperMap has offered products with powerful mapping capability, including unique value thematic map, graph thematic map, compound label thematic map, what-you-see-is-what-you-get displaying ability, and rich point symbol library, line shapes and region fill-up style, and so on. The 3D GIS of SuperMap Objects Java can project 2D map into scene to directly show the world to us.

Steps to add maps into scene:

Obtain the scene that is going to contain the map;
Call `Layer3Ds.add()` method to add the map to the scene as a 3D layer in map type;

**Syntax:**

```java
public Layer3D add(String dataName, Layer3DType layerType, Boolean addToHead)
```

```java
public Layer3D add(String dataName, Layer3DType layerType, Boolean addToHead, String layerName)
```

**Parameters:**

- `dataName` is the full directory of data corresponding to the map type layer.
- `layerType` is the type of the layer. If the added data name does not match the layer's type, then the adding is failure.
- `addToHead` decides if the new added 3D layer is put on the top of others.
- `layerName` is the name of the specific 3D layer, which is not case sensitive. If the layer name is already exist, then give abnormal.

**Return Value:**

Return the index value of the 3D tracking layer that contains the 3D geometric objects.

### 3.3 Thematic Maps

SuperMap Objects Java allows the 3D thematic map that is generated from 2D vector dataset to be added into the scene, outputting 3D label map, 3D ranges map, and 3D unique value map, for the detail instruction of thematic maps, please refer to the Thematic Maps in the Technical Documents.

**Steps to add 3D thematic maps into scene:**

1. Obtain the scene that is going to contain 3D thematic map;
2. Produce 3D thematic maps;
3. Call the method, and add the 3D thematic map to the scene as a 3D layer in dataset type;

**Syntax:**
Layer3DDataset Scene.addLayers().add(Dataset dataset, Theme3D theme3D, Boolean addToHead)

Parameters:

- dataset is the specific 2D vector data.
- theme3D is the specific 3D thematic map. Class Theme3D is the base class of the 3D thematic map.
- addToHead decides if the new added 3D layer is put on the top of others.

Return Value:

The 3D layer object in dataset type
Diagram 12 3D ranges map

Diagram 13 3D label map
4 Integration of Two and Three Dimensional Analysis

4.1 Surface Measurement

3D GIS of SuperMap Objects Java provides 3D measurement and many 2D GIS measurements with high precision.

(1) Surface distance measurement Surface distance measurement calculates the surface distance between two points or multiple points. There are two kinds of calculations, the arc distance and the topographical distance. When the terrain data is loaded, the result of calculation is topographical distance; else, the result of calculation is arc distance.

Since the earth is nearly a sphere, the distance between two points is the arc length of the circle centered as geocenter. The topographical distance is the actual distance between the two points, which involves the fluctuation of terrain. By default, it uses the arc distance, with meter as the unit.

(2) Surface area measurement

Whether the terrain data is loaded or not, the result of the surface measurement is the area of triangles which is consisted of the points by mouse clicking in the scene, with unit of square meter.

(3) Altitude measurement

Altitude distance is the elevation distance between two points. This method is very convenient for calculating the floors of a building or the height of a mountain.
4.2 Network Analysis

3D GIS of SuperMap Objects Java keeps the powerful 2D analysis functionality, including public transport transfer, cache analysis, path finding analysis, the closest facility query, service area analysis, etc. Please refer to the Network Analysis section in the Technical Documents.
4.3 Simple Query

The 3D GIS of SuperMap offers query functionality to obtain information of location, elevation, and any object's attributes. The objects are classified as 3D points, 3D lines, 3D regions, or as 3D geometric objects and 3D model objects, and so on.
Remarks: the elevation can be obtained from corresponding terrain data. In order to query elevation, the terrain data must be contained in the scene. If there is no terrain data, return -1, which means invalid elevation value.

### 4.4 SQL Query

The core GIS functionality for projects are spatial query and analysis. When using GIS data, we have to find some data that meets specific conditions from the existing data. In 3D GIS of SuperMap, it provides SQL query function. Please refer to the "SQL Query Documents" for detail.

SQL query indicates the query conditions only have relationships with spatial objects' attributes, not the geographical locations. For questions like "List the provinces of China that have a population larger than 50 million?", "List the counties in Hebei province have grain production that is more than one million?", "List lakes in Jiangsu province cover larger than 100 sq km?", "List the hotels in Beijing have more than 3 stars?" are suitable to be resolved by this method.

### 4.5 Spatial Query

Spatial query is a query method generating filler according to the spatial location relationship. It
also allows to apply the traditional attribute query method.

There are 9 basic operators and self-defined operator for calculations. The query outcome can be displayed in the attribute window, and the object in the scene can be located through Attributes. There are three factors are necessary for spatial query, the object, the target layer and the outcome recordset. Please refer to "Spatial Query Documents" for detail.
5

Basic Concepts in Scene

5.1 General Introductions of Three Dimensional Window and Scene

SuperMap Objects Java provides a 3D window class (SceneControl class). All the 3D visible operations should be carried out in this window, each of which can only display one scene (Scene class). According to the related workspace (Scene.Workspace attribute), a scene is sorted in a scene set (Workspace. Scenes attribute) of the workspace.

In a new created 3D window, a scene exists in the window by default. You may obtain current displaying scene of the 3D window by Scene attribute of the 3D window control (SceneControl class). Please look at the Diagram 17.

The main sector of the three dimensional scene in SuperMap Objects Java is a virtual three dimensional sphere (with a radius 6,378,137m) to model the earth. The sphere is marked by the latitude and longitude degree to provide geo-reference. Users are able to browse easily according to the latitude-longitude net of the scene. Meanwhile, by using the global remote sensing image as the setting, together with the environment factors of stars, the atmosphere, and the fog embracing the earth, users will receive a realistic earth. There are camera functionality to control the observation angle, position and scope to the earth.

The 3D window provides navigation tools, operation state interfaces, and browsing by mouse abilities to users, including zoom in, zoom out, incline, rotate and so on, and some simple operations like distance measurement, square measure, and height measurement.
The attributes of the three dimensional window class (SceneControl class) and the three dimensional scene class (Scene class) are illustrated in Table 4 and Table 5.

Table 4 Methods of SceneControl class

<table>
<thead>
<tr>
<th>METHODS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>get/setAction</td>
<td>Obtains or sets the 3D map operation state. Please refer to Action3D enumeration class about the map operation state.</td>
</tr>
<tr>
<td>is/setKeyboardNavigationEnabled</td>
<td>Obtains or sets if it is allowed to be browsed by keyboard.</td>
</tr>
<tr>
<td>is/setMouseNavigationEnabled</td>
<td>Obtains or sets if it is allowed be browsed by mouse.</td>
</tr>
<tr>
<td>is/setStatusBarVisible</td>
<td>Obtains or sets if the state bar is visible.</td>
</tr>
<tr>
<td>getNavigationControl</td>
<td>Obtains three dimensional map navigation objects.</td>
</tr>
</tbody>
</table>
**getScene**  
Obtains scene objects (Scene).

### Table 5 Methods of Scene Class

<table>
<thead>
<tr>
<th>PROPERTIES</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>get/setAtmosphere</td>
<td>Obtains or sets the atmosphere environment objects.</td>
</tr>
<tr>
<td>get/setBackgroundColor</td>
<td>Obtains or sets the setting color.</td>
</tr>
<tr>
<td>get/setCamera</td>
<td>Obtains or sets the camera objects of current scene.</td>
</tr>
<tr>
<td>get/setCameraFOV</td>
<td>Obtains or sets viewpoints of the scene camera, with a default value of 45 degree. The unit is degree.</td>
</tr>
<tr>
<td>get/setFog</td>
<td>Obtains or sets the fog objects. To render the 3D image by setting the Fog object in the scene.</td>
</tr>
<tr>
<td>get/setFogVisibleAltitude</td>
<td>Obtains or sets the visible degree, i.e. the elevation height.</td>
</tr>
<tr>
<td>is/setBackgroundImageVisible</td>
<td>Obtains or sets if the setting image is visible.</td>
</tr>
<tr>
<td>is/setScaleLegendVisible</td>
<td>Obtains or sets if the scale is visible, with &quot;True&quot; as visible, &quot;False&quot; as not.</td>
</tr>
<tr>
<td>getLatLonGrid</td>
<td>Obtains latitude-longitude net objects.</td>
</tr>
<tr>
<td>getLayers</td>
<td>Obtains 3D layer set objects.</td>
</tr>
<tr>
<td>get/setMaxCameraDistance</td>
<td>Obtains or sets the scene camera's longest distance value. The unit is meter.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>get/setMinCameraDistance</code></td>
<td>Obtains or sets the scene camera’s shortest distance value. The unit is meter.</td>
</tr>
<tr>
<td><code>get/setName</code></td>
<td>Obtains or sets the current scene name.</td>
</tr>
<tr>
<td><code>getScreenLayer</code></td>
<td>Obtains screen layers.</td>
</tr>
<tr>
<td><code>get/setTerrainExaggeration</code></td>
<td>Obtains or sets the scale of the terrain exaggeration.</td>
</tr>
<tr>
<td><code>getTerrainLayers</code></td>
<td>Obtains the terrain layer set of the 3D map scene.</td>
</tr>
<tr>
<td><code>getTrackingLayer</code></td>
<td>Obtains 3D tracking layer objects.</td>
</tr>
<tr>
<td><code>get/setWorkspace</code></td>
<td>Obtains or sets the workspaces related to the 3D map scene.</td>
</tr>
</tbody>
</table>

Besides using the stated default scene of the three dimensional window to conduct three dimensional applications, users can open an existing three dimensional scene or create a new scene object:

Open the existing scene

If there are scenes stored in the scene set of the workspace, you may open its related scene by the specified names, using `Open()` of the Scene class.

Syntax:

```java
Boolean Scene.open(String sceneName)
```

Create scene objects by using XML data

Open the scene by its XML data that describes the scene.

Syntax:

```java
Boolean Scene.from(String xml)
```

Create a new scene object

Syntax:
Scene sceneObject = new Scene();

Scene sceneObject = new Scene(Workspace workspace);///<Create a scene class object according to the specified workspace.

SuperMap Objects Java offers three dimensional window and scene as a framework to add various three dimensional data on this base. In other words, the three dimensional data, based on its geo-reference information, will be added on the 3D sphere in the SuperMap Objects Java through the interfaces offered by the scene (please refer to Table 5). The following paragraphs will introduce all the 3D data that could be added into scene in detail, and how to achieve it.

## 5.2 Camera

In SuperMap Objects Java 3D GIS, the camera object (Camera) is used to control the viewpoint of three dimensional scene, which works as a virtual camera lens to direct the view.

The attributes of the camera object are illustrated in the Table 6, by which you may control the position and direction of the camera:

<table>
<thead>
<tr>
<th>METHOD</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>get/setAltitude</td>
<td>Obtains or sets the height of the camera.</td>
</tr>
<tr>
<td>get/setAltitudeMode</td>
<td>Obtains or sets the altitude mode.</td>
</tr>
<tr>
<td>get/setHeading</td>
<td>Obtains or sets the azimuth angle (with the North direction). The scope of the azimuth angle is 0 - 360.</td>
</tr>
<tr>
<td>is/setEmpty</td>
<td>Obtains or sets if the camera object is null.</td>
</tr>
<tr>
<td>get/setLatitude</td>
<td>Obtains or sets the latitude degree of the camera. The unit is degree.</td>
</tr>
<tr>
<td>get/setLongitude</td>
<td>Obtains or sets the longitude degree of the camera. The unit is degree.</td>
</tr>
<tr>
<td>get/setTilt</td>
<td>Obtains or sets the elevation angle or depression angle of the camera. The scope of the depression angle is 0 - 90 degree.</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

Camera object can help us to know the answers of questions in Table 7:

**Table 7 What does the camera object tells us**

<table>
<thead>
<tr>
<th>QUESTIONS</th>
<th>ANSWERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where is the camera in the space?</td>
<td>A point located by attributes of Longitude, Latitude, Altitude as well as AltitudeMode. Generally speaking, the camera should not be put on the ground.</td>
</tr>
<tr>
<td>If the direction of the camera lens can make sure the north of the view directing upward?</td>
<td>If the north of the view is upward, the default value, 0, of Heading attribute will be used; otherwise, please rotate the camera.</td>
</tr>
<tr>
<td>If the camera lens is on the right top of the earth?</td>
<td>If the camera is on the top of the earth, the default value of attribute Tile is 0; otherwise, please rotate the camera around the &quot;X&quot; axis (The value of attribute Tilt could have negative value, which may cause upside down view).</td>
</tr>
</tbody>
</table>

SuperMap Objects Java 3D GIS applies the Cartesian Coordinates Datum to manage all the data, taking the earth as an ellipsoid. In order to convert between coordinate systems easily and rapidly, an earth-cylinder with a radius 6,378,137 m is defined in the 3D environment. The origin of the Cartesian Coordinates Datum is the center of the sphere; the Z axis is the crossed line of meridian plane and equator plane; the line that is on the equator plane and orthogonal with Z axis is the X axis; The Y axis is orthogonal with X axis and Z axis. This is the right handed Cartesian system; please look at the Diagram 18.

The axes X', Y' and Z' in Diagram 18 which are parallel with the Cartesian Coordinates Datum, is an coordinate system added on the camera. The grey solid line is the view direction from the camera.
to the center of sphere.

In default state, the camera is located on the crossed point between prime meridian and equator, with both attributes Heading and Tilt as zero. Hence, the changing of camera's direction and the position to get different view means to give new attribute values. The value of Heading indicates to rotate alone the Z axis, while the value of Tilt will rotate the camera alone the X axis.

Diagram 18 The relationship between the sphere coordinate system and camera in SuperMap Objects Java 3D GIS

X' axis directs to the right of the camera lens.

Y' axis defines the "upward" direction relative to the screen.

Z' axis directs to the viewpoint from the center of screen. The camera lens looks along the -Z' axis.

Table 8 The scene view with different camera parameters
<table>
<thead>
<tr>
<th>Altitude : 7500km</th>
<th>![Image 1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heading : 0</td>
<td>![Image 2]</td>
</tr>
<tr>
<td>Tilt : 0</td>
<td></td>
</tr>
<tr>
<td>Longitude : 0</td>
<td></td>
</tr>
<tr>
<td>Latitude : 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Altitude : 7500km</th>
<th>![Image 3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heading : 0</td>
<td>![Image 4]</td>
</tr>
<tr>
<td>Tilt : 0</td>
<td></td>
</tr>
<tr>
<td>Longitude : 126</td>
<td></td>
</tr>
<tr>
<td>Latitude : 23</td>
<td></td>
</tr>
</tbody>
</table>

*SuperMap Objects Java Technology Documents*
5.3 Altitude Mode

The 3D GIS in SuperMap Objects Java offers three latitude mode types, ClampToGround, RelativeToGround, and the Absolute. The latitude mode is mainly used to direct the 3D GIS of SuperMap Objects Java to analyze the elevation value of the 3D data. The following paragraphs are introductions of the explicit meaning of each latitude type.

**ClampToGround**: In this latitude type, the elevation of 3D data will be ignored. According to the latitude-longitude information and the terrain shape, the 3D objects will be just on the terrain surface. Please look at Diagram 19, add a three-dimensional region object into a scene with terrain data (into a KML layer actually). Under the ClampToGround mode, the region is displayed along with the terrain wave.

The latitude mode type is the default mode in 3D GIS of SuperMap Objects Java.
Absolute: The elevation in the absolute latitude mode is recorded from sea level, which will ignore terrain's elevations. The absolute latitude is very useful when the precise elevation values are available. For instance, GPS is able to trace the route of flying and diving under Absolute mode. Please look at Diagram 20, add a three-dimensional region geometric object (GeoRegion3D class object, the green part in Diagram 20), into a scene environment (into a KML layer actually). The boundary elevation value of the region is 5800m; as a result it is displayed as the image in Diagram 20 under the Absolute mode.
RelativeToGround: The elevation value in this mode is recorded according to the earth surface. For instance, to build wire poles, with a length 25 meter, in mountain area. Under theRelativeToGround mode, as Diagram 21 shows, each elevation of the top of pole will go along with the terrain wave; the yellow vertical lines are wire poles, and the horizontal lines are wires.
5.4 Texture Mapping

The 3D GIS in SuperMap Objects Java are able to give texture map to the three dimensional objects. Additionally, the top, medium, and bottom parts could have different texture maps to reflect the realistic object.

Diagram 22 and Diagram 23 display the effect of a three dimensional geometric object before and after being processed by a texture map.
SuperMap Objects Java requires the length and width of the images that are used as texture map should be multiple of two, i.e. 2\text{n}. Otherwise, the image should be re-sampled. In order to get transparent effect, the format of the texture map should be in RGBA. 3D GIS in SuperMap Objects Java allows pictures of *.bmp, *.gif, *.png, *.jpg, *.jpeg, *.tga, *.tif, *.tiff.

When the dimensions of the texture map are smaller than the object, users can choose to repeat mapping or stretch the texture map. The texture mapping repetition will keep the original resolution rate, giving clear effect; however, some texture pattern is not nature on the transition.
position. The stretched texture map will produce evenly but indistinct effect. Users should select a suitable one to illustrate the objects.

Remarks: any two texture images should not have the same name, even which are in different directories. When stretching a texture image, it is not allowed to have transparent effect. In order to ensure the system performance, it is recommended to have texture image within 1024*1024.
Three Dimensional Operations

The 3D operations of 3D GIS of SuperMap Objects Java only provides 3D browsing function, including zooming in and out, incline, pitch, roll, and so on. Users are allowed to do the browsing operations by 3D navigation tool offered by 3D window, the combination of mouse and keyboard, as well as the interfaces.
The following table lists frequently used browsing operations, interfaces, mouse and keyboard operations, and navigation tool.

Table 9 Browsing functionality operated by mouse and keyboard and the corresponding navigation tool
<table>
<thead>
<tr>
<th>BROWSING FUNCTIONS</th>
<th>CORRESPONDENT INTERFACES</th>
<th>MOUSE OPERATIONS</th>
<th>KEYBOARD OPERATIONS</th>
<th>NAVIGATION TOOL BAR</th>
</tr>
</thead>
</table>
| Browse              | Scene. pan (Double offsetLongitude,
Double offsetLatitude)
Method            | Left click and drag   | Cursor movement
keys(up, down, left, right) |                      |
| Zoom In             | Scene. zoom (Double ratio)
Method                  | Use the middle roll
or right click and drag
up / down.                  | PageUp and PageDown          |                      |
| Incline of scene    | Scene. pitch (Double angle)
Method                  | Middle click and drag up
/ down                      | Shift + up / down cursor
key                       |                      |
<table>
<thead>
<tr>
<th>Rotate along the scene center</th>
<th>Scene. roll (Double angle) Method</th>
<th>Middle click and drag left / right</th>
<th>Shift + left / right cursor key</th>
</tr>
</thead>
</table>

![Roll Method](image)