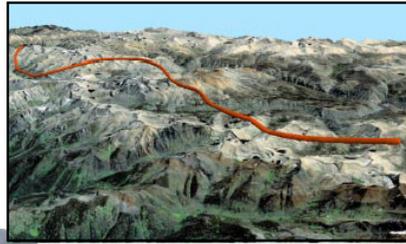
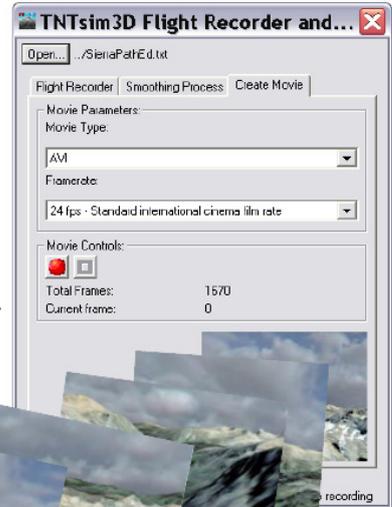


# 動画の作成

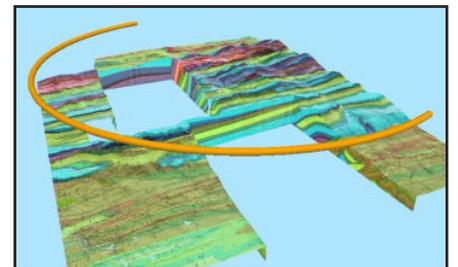
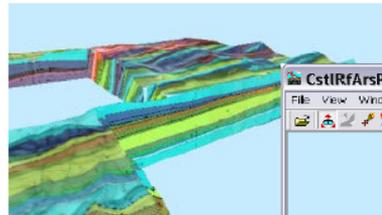
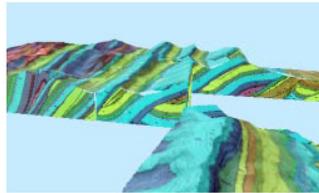
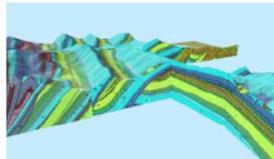


TNTsim3Dのフライトレコーダースクリプトを使って、飛行経路を記録し、これを用いて動画を作ることができます。

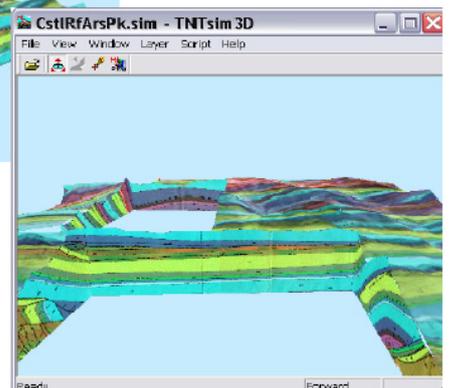


TNTsim3Dでは、自分の地理空間データを使ってリアルタイムの3次元シミュレーション上を自在に動くことができるだけでなく、それを配布して他の人でも標準のメディアプレーヤーで再生できる動画として記録することができます。動画の作成機能はフライトレコーダースクリプト（反対側のページ参照）によって実現されています。

フライトレコーダースクリプトの基本的な機能には、3次元景観の飛行経路の記録、再生、保存、再開などがあります。スクリプトから作られる対話画面を使えばこれらの動作を簡単に操作することができます。自分の目的に合った飛行経路ができれば、自由に経路をスムージングしたり、AVIまたはMPEG どちらかのフォーマットで動画を記録したりすることができます。経路のスムージングはカラー版“TNTsim3D:Smoothing Flight Paths”（飛行経路のスムージング）で解説しています。動画の記録中、TNTsim3Dのメイン画面にレンダリングされる各フレームのスナップショットを撮って、選択したフレームレートで指定した動画ファイルに転送されます。動画のスピードをはじめのシミュレーションのスピードに合わせるには、希望する動画のフレームレートと等しいフレームレート（TNTsim3DのOptionsウィンドウの地形（terrain）パネルにおいて）を使うためにTNTsim3Dの設定の最初の飛行経路を記録してください。動画を作る際、そのように合わせた動画のフレームレートを選択します。飛行経路を再生してフレームを記録する際は、最高画質のレンダリングを使用するようにTNTsim3Dを設定します。（フレームレートを一定にするよりもこちらのほうが好ましいです。）レンダリングのスピードは動画を記録すると変化しますが、地形ノイズの少ない高画質の画像が設定したフレームレートで再生されます。



フライトレコーダーでは、他のスクリプトを使って計算した円軌道の一部のような飛行経路（テキストファイル）を用いて動画を作ることができます。



www.microimages.com/freestuf/scripts.htm にはダウンロード可能な多くのサンプルスクリプトがあり、スクリプトやクエリーで TNT 製品のスクリプト言語をどのように利用したらよいか解説しています。

## Advanced Flight Recorder スクリプト (抜粋) (AdvRecorder.sml)

### 動画記録の開始と記録パラメータを設定する処理

```
proc StartMovie() {
  TNTsim3D.StartRecording();
  frameDevice = TNTsim3D.GetFrameDevice();
  movieWidth = frameDevice.GetWidth();
  movieHeight = frameDevice.GetHeight();
  movieFrame = FrameCreate(movieWidth, movieHeight);
  gc = FrameCreateGC(movieFrame);
  ActivateGC(gc);
  movie = MovieInit();
  SetColor(color);
  data = dlgwin.GetValues();
  local string type$ = data.GetValueStr("MovieTypeCombo");
  local string framerate$ = data.GetValueStr("MovieFramerateCombo");
  print(type$);
  MovieSetFormat(movie, type$);
  print(framerate$);
  MovieSetFrameRate(movie, framerate$);
  MovieSetFrameWidth(movie, movieWidth);
  MovieSetFrameHeight(movie, movieHeight);

  string ext$;
  ext$ = MovieGetFileExt(movie);
  string filename$;
  filename$ = GetOutputFileName("", "Create output file for movie", ext$);
  printf("Filename = %s\n", filename$);
  MovieStart(movie, filename$);
  movierect.Set(0,0, movieWidth-1, movieHeight-1);
  moviepoint.x = 1;
  moviepoint.y = 1;
  SetState(3);
  moviestop.SetEnabled(1);
}
```

出力ファイルの作成

コンボボックスからタイプを使用

コンボボックスから  
フレームレートを使用

### 動画記録の終了とクリーンアップ処理

```
proc EndMovie() {
  MovieStop(movie);
  DestroyGC(gc);
  MovieExit(movie);
  FrameDestroy(movieFrame);
  TNTsim3D.StopRecording();

  moviestop.SetEnabled(false);
  moviestop.SetValue(0,0);
  movierecord.SetValue(0,0);
  print("END");
  state = 0; # Don't use SetState() here since it would call EndMovie again
  index = 0;
}
```

ボタンをグレー表示する

押されたらストップボタンを出す

レコードボタンを出す

TNTsim3D であらかじめ定義された SML の関数名で、各フレーム (コマ) ごとに呼び出されます。レコーダーの状態によって指定された通りに飛行中のフレームを記録、再生します。

```
proc OnFrame () {
  if (state == 1) {
    local class POINT3D viewer, orientation;
    TNTsim3D.GetSceneByOrientation(viewer, orientation);
    viewer.x[index] = viewer.x;
    viewery[index] = viewer.y;
    viewerz[index] = viewer.z;
    orientation.x[index] = orientation.x;
    orientation.y[index] = orientation.y;
    orientation.z[index] = orientation.z;
    numberframes.SetLabel(NumToStr(numindices));
  }
```

記録している時、現在の視点の位置や向きを得て配列に保存します。

視点の位置

視点の向き (x=ピッチ (伏角)、y=ロール (回転)、z=機首方位)

ダイアログ中のフレームカウンタ  
ベルを更新

```
movietotalframes.SetLabel(NumToStr(numindices));
frame.SetLabel(NumToStr(index));
index++;
numindices++;
if (numindices >= maxindices)
  OnStop();
}
```

カウントを増やす

再生する場合、保存された視点の位置や向きを配列から読み込み、現在のフレームの設定に使う。

```
else if (state == 2) {
  local class POINT3D viewer, orientation;
  viewer.x = viewerx[index];
  viewer.y = viewery[index];
  viewer.z = viewerz[index];
  orientation.x = orientationx[index];
  orientation.y = orientationy[index];
  orientation.z = orientationz[index];
  TNTsim3D.SetSceneByOrientation(viewer, orientation);
  frame.SetLabel(NumToStr(index));
  index++;
  if (index >= numindices) {
    if (loop == 1)
      index = 0;
    else OnStop();
  }
}
```

視点の位置

視点の向き (x=ピッチ (伏角)、y=ロール (回転)、z=機首方位)

フレームを設定

ダイアログ中のフレーム  
カウンタベルを更新

フレームカウントを増やす

[ループ再生 (Loop)] チェックボックスがオンのとき、ストップするまで再生を繰り返します。

動画を生成中の場合、保存された視点の位置や向きを配列から読み込み、現在のフレームにセットします。フレームをコピーして現在の動画中に入れます。

```
else if (state == 3) {
  print(index);
  moviecurrentframe.SetLabel(NumToStr(index));
  local class POINT3D viewer, orientation;
  viewer.x = viewerx[index];
  viewer.y = viewery[index];
  viewer.z = viewerz[index];
  orientation.x = orientationx[index];
  orientation.y = orientationy[index];
  orientation.z = orientationz[index];
  TNTsim3D.SetSceneByOrientation(viewer, orientation);
  frame.SetLabel(NumToStr(index));
  index++;
}
```

視点の位置

視点の向き (x=ピッチ (伏角)、y=ロール (回転)、z=機首方位)

フレームを設定

ダイアログ中のフレーム  
カウンタベルを更新

デバイスが正しいサイズかどうかを確認します。

```
if (frameDevice.GetHeight() != movieHeight
  || frameDevice.GetWidth() != movieWidth) {
  PopupMessage("Window Resize Detected. Movie generation stopped.");
  EndMovie();
  return;
}
```

長方形の画像をメモリ中のグラフィックスデバイスから、現在の動画フレームにコピーします。

```
gc.CopyRect(frameDevice, movierect, moviepoint);
MovieAddFrame(movie, movieFrame);
```

フレームを動画に加えます。

```
if (index >= numindices)
  EndMovie();
}
```