

5. Dựng Video từ đối tượng 3D với POV-Ray



5.1 Cài đặt và nạp thư viện

Cài đặt thư viện

```
1 !pip install vapory
2 !apt-get update
3 !apt-get install povray
4 !pip install scikit-image
```

Nạp thư viện

```

1 from vapory import *
2 from moviepy.editor import VideoClip
3 from skimage.filters import sobel
4 import numpy as np

```

5.2 Triển khai POV-Ray với đối tượng 3D trong Python

5.2.1 Render một hình tròn 3D đơn giản

Khởi tạo đối tượng:

```

1 camera = Camera( 'location', [0,2,-3], 'look_at', [0,1,2] ) #Thiết lập camera
2 light = LightSource( [2,4,-3], 'color', [1,1,1] ) #Thiết lập nguồn sáng
3 sphere = Sphere( [0,1,2], 2, Texture( Pigment( 'color', [1,0,1] ))) #Khởi tạo đối tượng

```

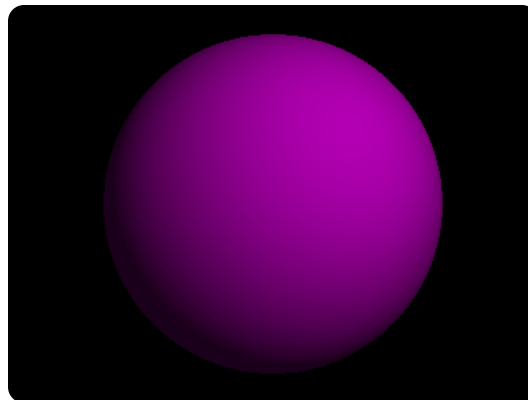
Render đối tượng:

```

1 scene = Scene( camera, objects= [light, sphere])
2 scene.render("purple_sphere.png", width=400, height=300)

```

Kết quả:



5.2.2 Post-preprocessing

Tạo cảnh trong đó vị trí của các vật thể phụ thuộc vào thời gian:

```

1 color = lambda col: Texture( Pigment( 'color', col))
2 def scene(t):
3     """ Returns the scene at time 't' (in seconds) """
4     return Scene( Camera( 'location', [0, 2, -3], 'look_at', [1, 1, 2] ),
5                   [ LightSource( [2, 4, -3], 'color', [1.5,1.5,1.5] ),
6                     Background( "color", [1,1,1] ),
7                     Sphere( [0, 1, 2] , 2, color([.8, 1, .2])),
8                     Box( [-.8 + .5 * t, -1.5, -.5] , [-.75+.5*t, 3.5, 5], # <= t
9                           color([1,.6,.5]), 'rotate', [0, 30, 0] ),

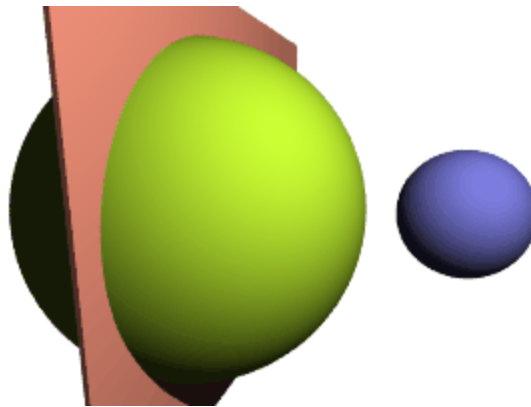
```

```
10 Sphere( [ 3 - 2 * t , 1, 1.1] , .75, color([.5, .5, .9]))])
```

Render đối tượng

```
1 def make_frame(t):
2     return scene(t).render(width = 300, height=200, antialiasing=0.001)
3 VideoClip(make_frame, duration=4).write_gif("anim.gif",fps=20)
```

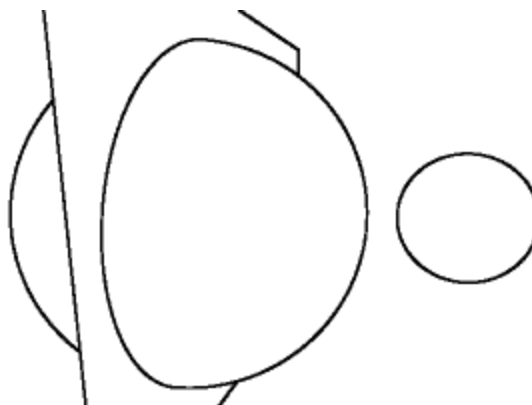
Kết quả:



Xử lý hậu kỳ với Sobel

```
1 def make_frame(t):
2     # We will use "quality=1" so that shadows won't be rendered,
3     # and double the rendering resolution to avoid pixelization.
4     im= scene(t).render(width = 600, height=400,
5                           antialiasing=0.001, quality=1)
6     sobelized = np.array([sobel(1.0 * im[:, :, i]) for i in [0, 1, 2]])
7     return np.dstack(3*[255*(sobelized.max(axis=0)==0)])
8
9 clip = VideoClip(make_frame, duration=4).resize(0.5)
10 clip.write_gif("anim_sobel.gif",fps=20)
```

Kết quả:



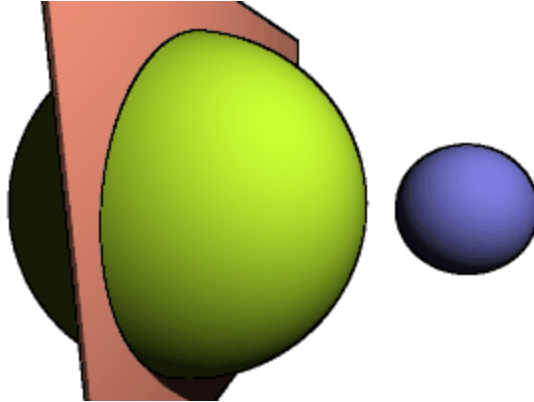
Kết hợp hai hình ảnh động để tạo hiệu ứng đổ bóng:

```

1 normal = VideoFileClip("anim.gif") # The first animation
2 sobelized = VideoFileClip("anim_sobel.gif") # The second animation
3 cel_shade = lambda gf, t: np.minimum(gf(t), sobelized.get_frame(t))
4 normal.fl(cel_shade).write_gif("cel_shaded.gif")

```

Kết quả:



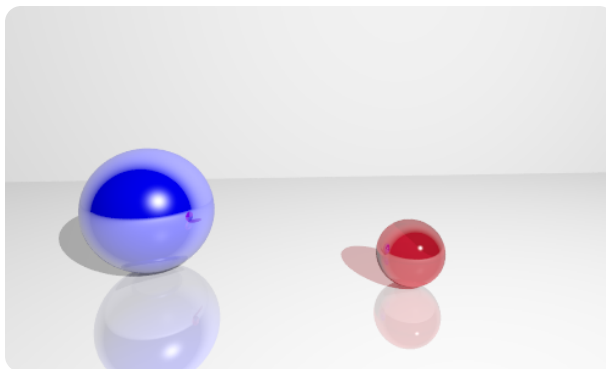
5.2.3 Nhúng video vào cảnh 3D

Khởi tạo đối tượng

```

1 light = LightSource([10, 15, -20], [1.3, 1.3, 1.3])
2 wall = Plane([0, 0, 1], 20, Texture(Pigment('color', [1, 1, 1])))
3 ground = Plane([0, 1, 0], 0, Texture(Pigment('color', [1, 1, 1]), Finish('phong', 0.1,
4                                     'reflection', 0.4, 'metallic', 0.3)))
5 sphere1 = Sphere([-4, 2, 2], 2.0, Pigment('color', [0, 0, 1]), Finish('phong', 0.8, 'reflection', 0.5))
6 sphere2 = Sphere([4, 1, 0], 1.0, Texture('T_Ruby_Glass'), Interior('ior', 2))
7
8 scene = Scene(Camera("location", [0, 5, -10], "look_at", [1, 3, 0]), objects = [ground, wall, sphere1, sph
9               included=["glass.inc"] )

```



Tải video bunny: [bunny.mp4](#)

Chèn Video vào đối tượng:

```

1 def embed_in_scene(image):
2

```

```

3     ffmpeg_write_image("__temp__.png", image)
4     image_ratio = 1.0*image.shape[1]/image.shape[0]
5     screen = Box([0, 0, 0], [1, 1, 0], Texture(
6         Pigment( ImageMap('png', '"__temp__.png"', 'once')),
7         Finish('ambient', 1.2) ),
8         'scale', [10, 10/image_ratio,1],
9         'rotate', [0, 20, 0],
10        'translate', [-3, 1, 3])
11     new_scene = scene.add_objects([screen])
12     return new_scene.render(width=800, height=480, antialiasing=0.001)
13
14 clip = (VideoFileClip("bunny.mp4") # File containing the original video
15         .subclip(23, 47) # cut between t=23 and 47 seconds
16         .fl_image(embed_in_scene) # <= The magic happens
17         .fadein(1).fadeout(1)
18         .audio_fadein(1).audio_fadeout(1))
19 clip.write_videofile("bunny2.mp4", bitrate='8000k')

```

Kết quả:

Embedded Bick Buck Bunny - MoviePy + Vapory



BÀI TẬP 1: Khởi tạo đối tượng hình cầu với POV-Ray; thực hiện các yêu cầu bên dưới:

1. Thiết lập tọa độ x, y, z lần lượt là:

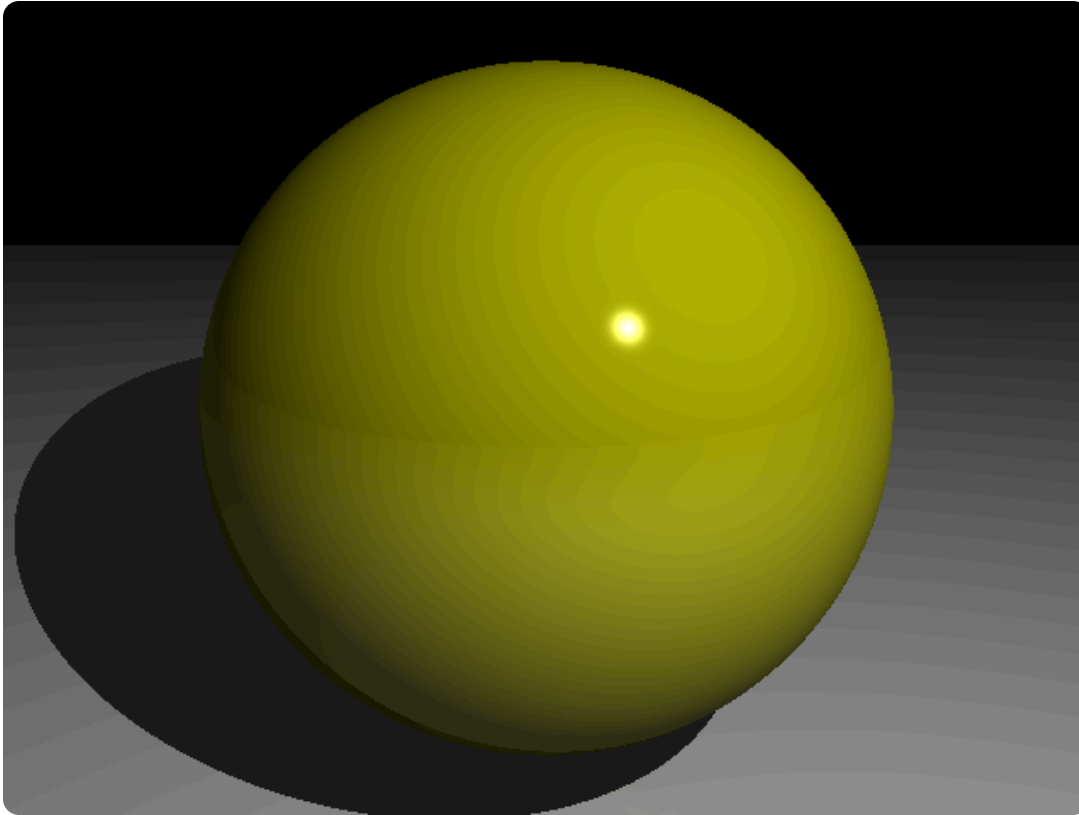
- x = [1, 0, 0]
- y = [0, 1, 0]
- z = [0, 0, 1]

2. Thiết lập camera biết rằng: 'location' là [0.0, 2.0, -3.0], 'up' là [0.0, 1.0, +0.0], 'right' là [4/3, 0.0, +0.0], 'look_at' là [0.0, 1.0, +2.0])

3. Thiết lập LightSource là [2, 4, -3] với 'color' là [1, 1, 1])

4. Thiết lập đối tượng Sphere là [0, 1, 2], 2.0, Texture(Pigment('color', [1, 1, 0]), Finish('phong', 1,'phong_size', 300, 'reflection', 0.15)))

5. Thiết lập Plane với `Plane(y,-1.5, Texture(Pigment('color', [1, 1, 1]), Finish('reflection', 0.10)))`
6. Compile đối tượng sử dụng phương thức Scene cho camera và `objects=[light, sphere, plane]`
7. Triển khai render đối tượng với tên `yellow_vapory.png` với width và height là 640 và 480
8. Kết quả



BÀI TẬP 2: Triển khai render các ví dụ bên dưới

```
1 x = [1, 0, 0]
2 y = [0, 1, 0]
3 z = [0, 0, 1]
4
5 #Thiết lập camera
6 camera = Camera('location', [1.65, 5.5, -5.0],
7                 'up',      [0.0, 1.0, 0.0],
8                 'right',   [4/3, 0.0, 0.0],
9                 'look_at', [0.0, 0.5, -1.0])
10
11 #Thiết lập ánh sáng
12 light1 = LightSource([-30, 11, 20], 'color', 'White')
13
14 light2 = LightSource([31, 12, -20], 'color', 'White')
15
16 light3 = LightSource([32, 11, -20], 'color', 'LightGray')
17
```

```

18 #Thiết lập kích thước khối lập phương
19 VEL = 1.45
20
21 box = Box([-VEL, -VEL, -VEL],
22           [VEL, VEL, VEL],
23           Texture(
24               'T_Wood23',
25               Finish('phong', 1,
26                     'phong_size', 300,
27                     'reflection', 0.15)))
28 plane = Plane(y,
29               -1.5,
30               Texture(
31                   'T_Stone1',
32                   Pigment('octaves', 3,
33                           'rotate', [i * 90 for i in z]),
34                   Finish('reflection', 0.10)))
35
36
37 def construct_scene(t):
38     sphere = Sphere([0, 3.3 - 7.0*t, 0],
39                    1.8,
40                    Texture(
41                        'T_Wood24',
42                        Finish('phong', 1,
43                              'phong_size', 300,
44                              'reflection', 0.15)))
45
46     csg_object = Difference(box, sphere)
47
48     # složení scény z jednotlivých objektů
49     return Scene(camera,
50                 objects=[light1, light2, light3, csg_object, plane],
51                 included=["colors.inc", "stones.inc", "woods.inc"],
52                 global_settings=["assumed_gamma 2.2"])
53
54
55 # Thiết lập tham số
56 DURATION = 10
57 FPS = 20
58
59
60 def make_frame(t):
61     scene = construct_scene(t / DURATION)
62     return scene.render(width=400, height=300, antialiasing=0.001)
63
64
65 animation = VideoClip(make_frame, duration=DURATION)
66 animation.write_videofile('scene4.ogv', fps=FPS, progress_bar=True, bitrate="800000")

```

BÀI TẬP 3: Triển khai render các ví dụ bên dưới

```

1 x = [1, 0, 0]
2 y = [0, 1, 0]
3 z = [0, 0, 1]
4
5 camera = Camera('location', [0, 20, -15],
6                 'look_at', [0, -2, 0])
7
8 light1 = LightSource([-50, 100, -80],
9                      'color', [1.0, 1.0, 1.0])
10
11 light2 = LightSource([250, 25, -100],
12                     'color', [0.85, 0.53, 0.10])
13
14 plane = Plane(y,
15              -6,
16              Texture(
17                  Pigment('checker',
18                          'color', 'Gray',
19                          'color', 'White*0.9'),
20                  Finish('reflection', 0.10),
21                  'scale', 4))
22
23 def create_torus(t):
24     return Torus(7.0, 4.0,
25                 Texture(
26                     Pigment('bozo',
27                             ColorMap([0.0, 0.4, 'color', [0.36, 0.20, 0.09],
28                                     'color', [0.36, 0.20, 0.09]],
29                                     [0.4, 1.01, 'color', [0.85, 0.57, 0.44],
30                                     'color', [0.85, 0.57, 0.44]]),
31                             'scale', [4, 0.15, 0.15],
32                             'rotate', [0, 45, 0])),
33                 Texture(
34                     Pigment('wood',
35                             'turbulence', 0.025,
36                             # čtyři barvy, které se ve vzorku střídají
37                             ColorMap([0.00, 0.15, 'color', [0.42, 0.26, 0.15],
38                                     'color', [0.85, 0.53, 0.10]],
39                                     [0.15, 0.40, 'color', [0.85, 0.53, 0.10],
40                                     'color', [1.00, 1.00, 1.00, 1.00]],
41                                     [0.40, 0.80, 'color', [1.00, 1.00, 1.00, 1.00],
42                                     'color', [0.85, 0.53, 0.10]],
43                                     [0.80, 1.01, 'color', [0.85, 0.53, 0.10],
44                                     'color', [0.42, 0.26, 0.15]]),
45                             'scale', [3.5, 1, 1],

```



```

46         'translate', [0, -50, 0],
47         'rotate', [0, 0, 10.0 * t]
48     ),
49     Finish('phong', 1,
50           'phong_size', 100,
51           'brilliance', 3,
52           'ambient', 0.2,
53           'diffuse', 0.8)))
54
55
56 def construct_scene(t):
57     torus = create_torus(t)
58     return Scene(camera, objects=[light1, light2, plane, torus],
59                 included=['colors.inc'],
60                 global_settings=['assumed_gamma 2.2'])
61
62 DURATION = 10
63 FPS = 20
64
65 def make_frame(t):
66     scene = construct_scene(t / DURATION)
67     return scene.render(width=400, height=300, antialiasing=0.001)
68
69 animation = VideoClip(make_frame, duration=DURATION)
70
71 animation.write_videofile('scene5.ogv', fps=FPS, progress_bar=True, bitrate="800000")

```

BÀI TẬP 4: Xây dựng đối tượng 3D và render với các yêu cầu sau:

1. import thư viện

```

1 from vapory import *
2 from moviepy.editor import VideoClip
3 from math import *

```

2. Thiết lập tọa độ x = 1, y = 1 và z = 1

3. Thiết lập camera là 'orthographic', 'location' là (0, 0, -1), 'right' là '5*4/3*x', 'up' là 'y*5', 'look_at' là (0, 0, 0)

4. Thiết lập 2 nguồn sáng: light 1: (2, 10, -10) với 'color' là (0.7, 0.7, 0.7) và light 2: (0, 0, -10000) với 'color' là (0.7, 0.7, 0.7), 'shadowless'

5. Thiết lập đối tượng Plane:

```

1 plane = Plane(z, 2, 'hollow', 'on', Texture(Pigment('agate', 'agate_turb', 0.9),
2         Finish('ambient', 0.1, 'diffuse', 0.4)))

```

5. Xây dựng hàm new_blob

```

1 def new_blob(threshold, dx, dy, t):
2     delta = 0.3 * sin(2*pi*t)
3     return Blob('threshold', threshold,
4                 'component', 1.0, 1.0, (0, -0.5 - delta, 0),
5                 'component', 1.0, 1.0, (0, 0.5 + delta, 0),
6                 Texture(
7                     Pigment('color', (0.6, 0.8, 1.0)),
8                     Finish('ambient', 0.2,
9                             'diffuse', 0.4,
10                            'phong', 0.5,
11                            'phong_size', 5)),
12                 'translate', (dx, dy, 0))

```

6. Xây dựng hàm construct scene:

```

1 def construct_scene(t):
2     objects = [light1, light2, plane]
3     objects.append(new_blob(0.40, -2.00, -1.2, t))
4     objects.append(new_blob(0.50, -0.65, -1.2, t))
5     objects.append(new_blob(0.60, 0.65, -1.2, t))
6     objects.append(new_blob(0.70, 2.00, -1.2, t))
7     objects.append(new_blob(0.75, -2.00, 1.2, t))
8     objects.append(new_blob(0.80, -0.65, 1.2, t))
9     objects.append(new_blob(0.82, 0.65, 1.2, t))
10    objects.append(new_blob(0.90, 2.00, 1.2, t))
11    return Scene(camera, objects=objects,
12                included=('colors.inc',),
13                global_settings=('assumed_gamma 2.2',))

```

7. Thiết lập tham số: FPS là 20, Duration là 15

8. Xây dựng hàm make_frame và triển khai render

```

1 def make_frame(t):
2     scene = construct_scene(t / DURATION)
3     return scene.render(width=400, height=300, antialiasing=0.001, auto_camera_angle=False)
4 animation = VideoClip(make_frame, duration=DURATION)
5 animation.write_videofile('scene8.ogv', fps=FPS, progress_bar=True, bitrate="800000")

```