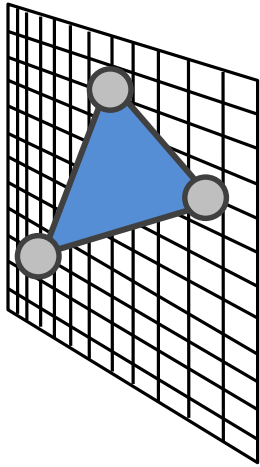


Computer Graphics (COMP0027) 2022/23

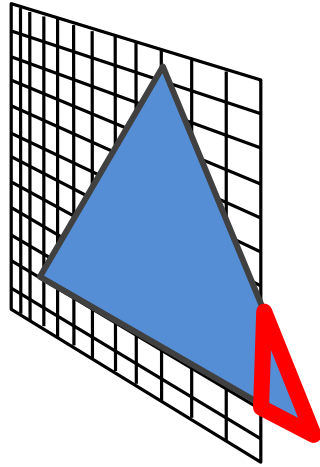
Shadows

Tobias Ritschel

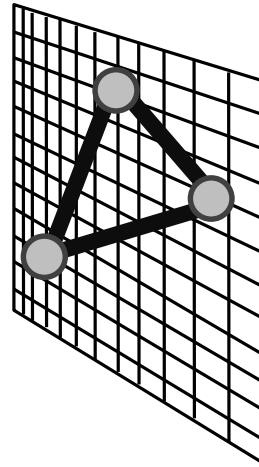
Challenges



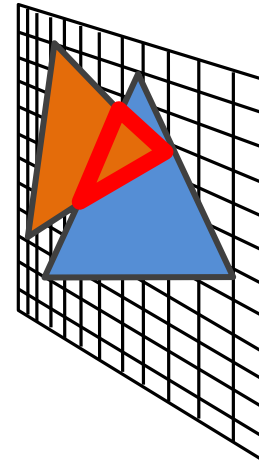
Projection



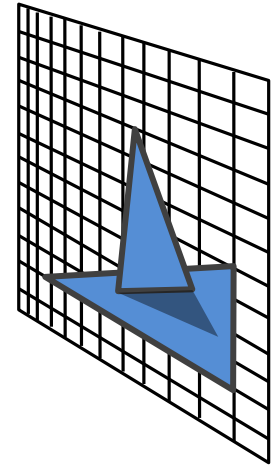
Clipping



Rasterization

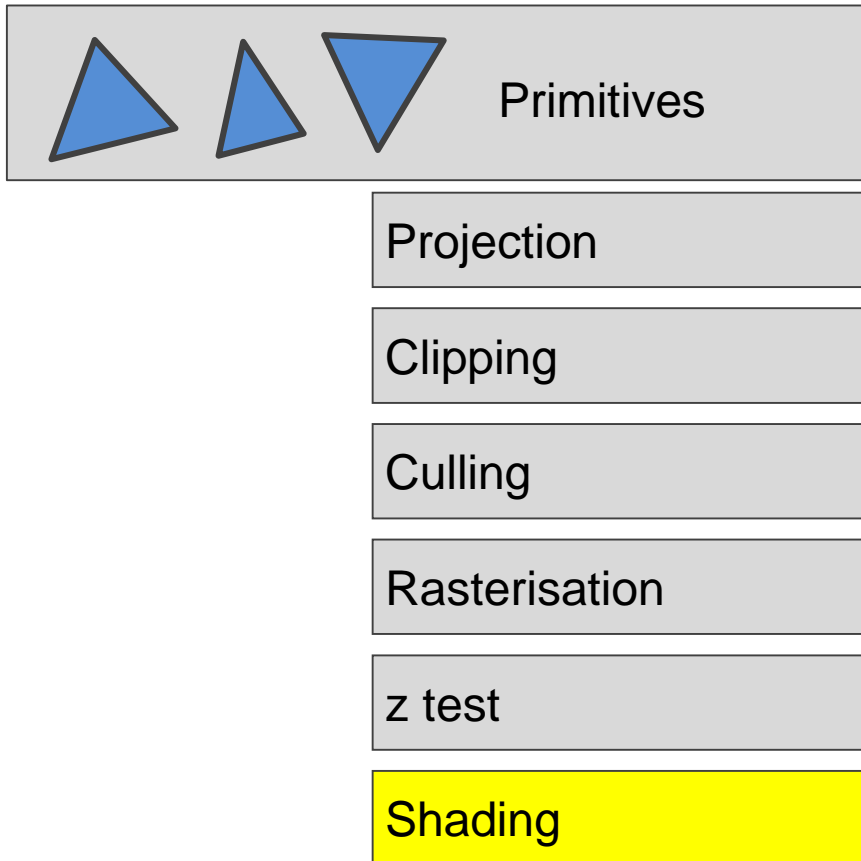


Visibility



Shading

Pipeline

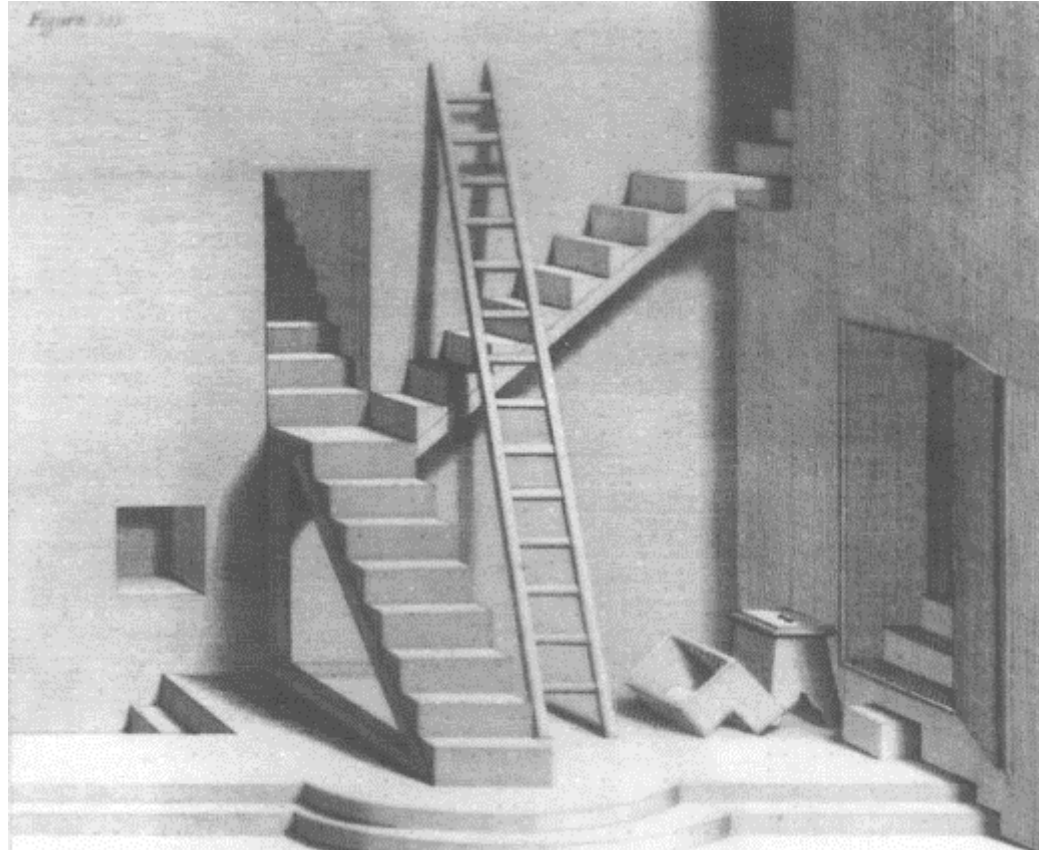


Outline

- Introduction
- Sharp shadows
- Soft shadows
- Conclusion

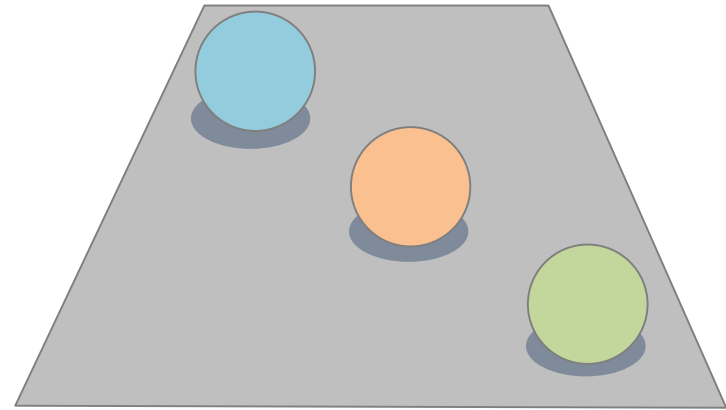
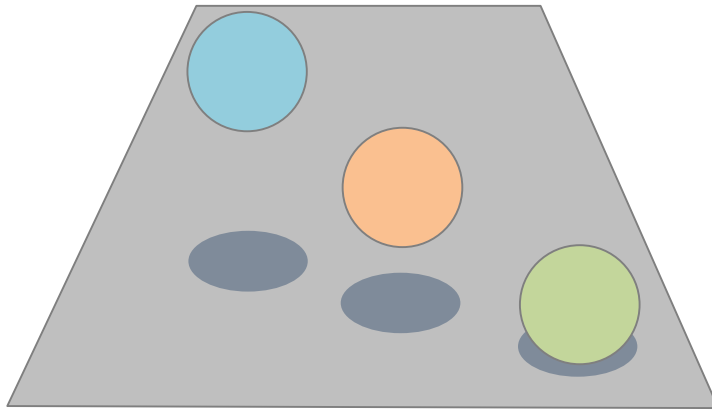
Why are Shadows Important?

- Depth cue
- Scene Lighting
- Realism
- Contact points

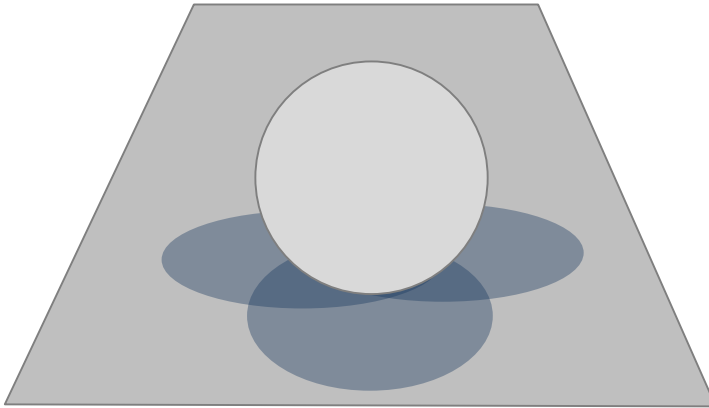


Spatial cue

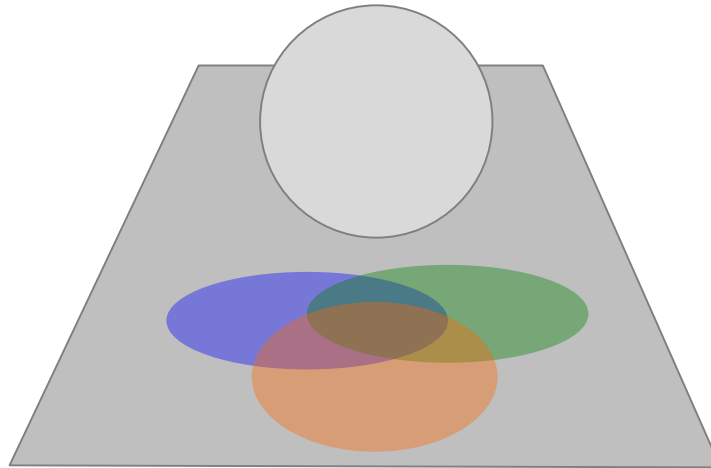
空间线索



Light position cue (sundial)



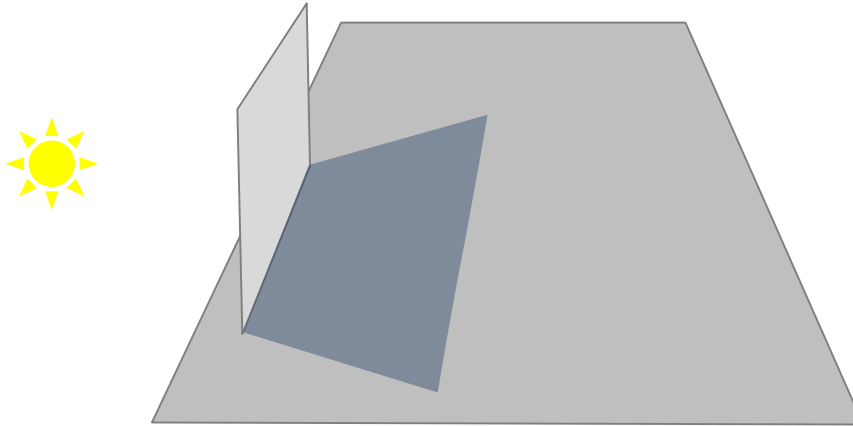
Light color



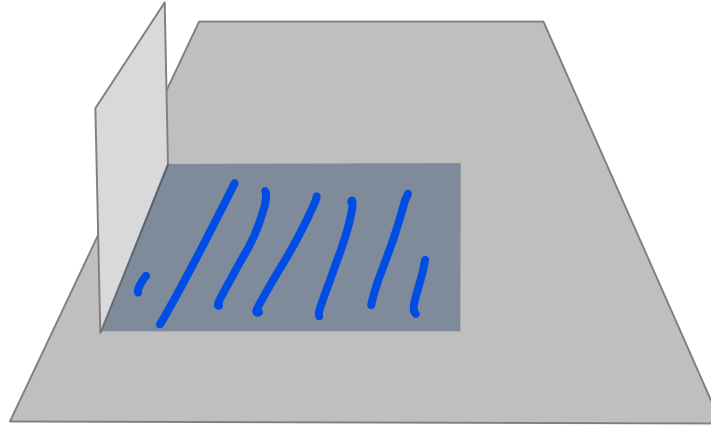
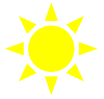


← shadow blue
是因为
天空是 blue 的
($K_{ambient} = blue$)
↑

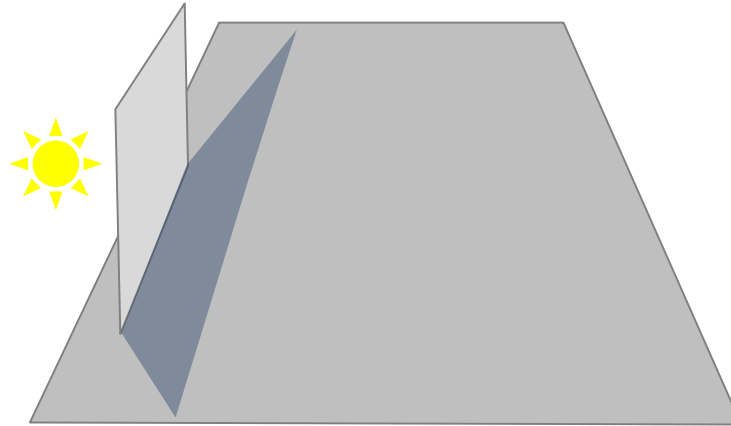
Light distance



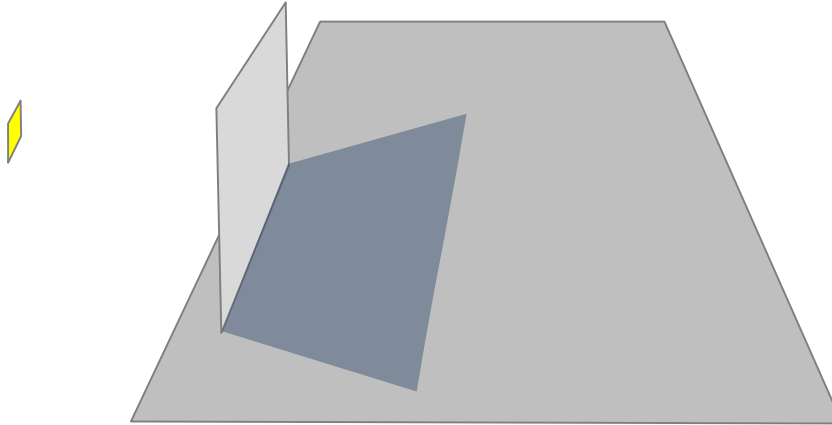
Light distance (far)



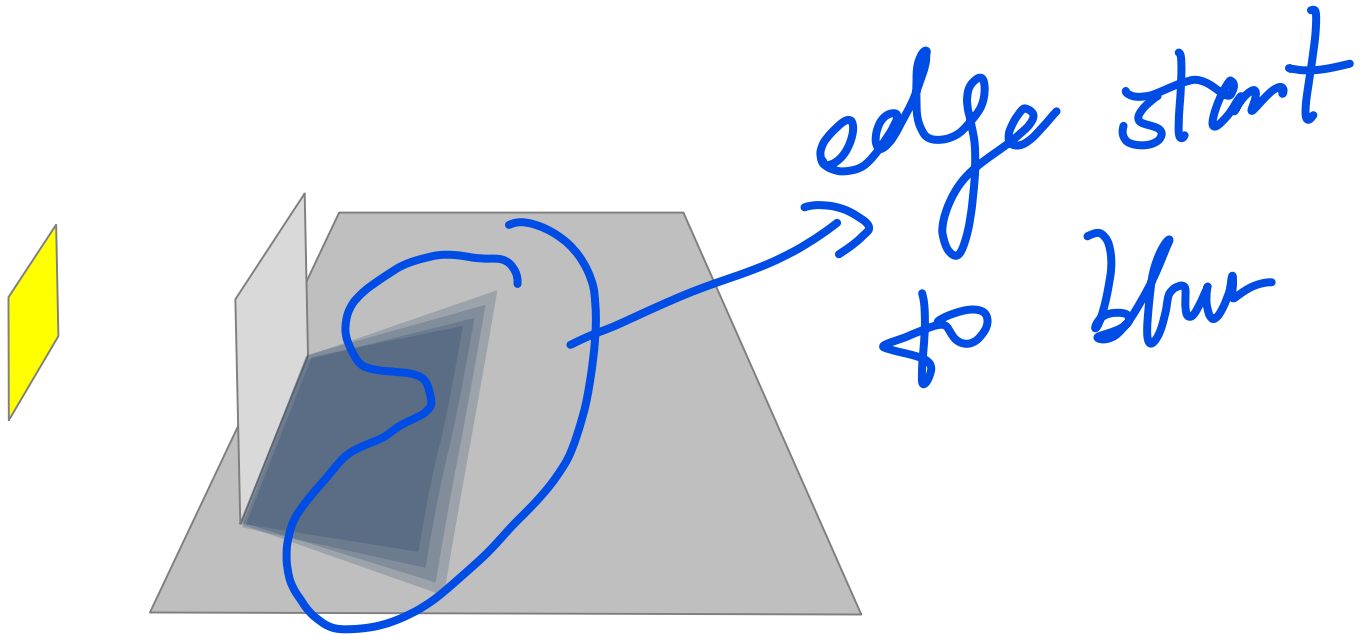
Light distance (near)



Light size (small)

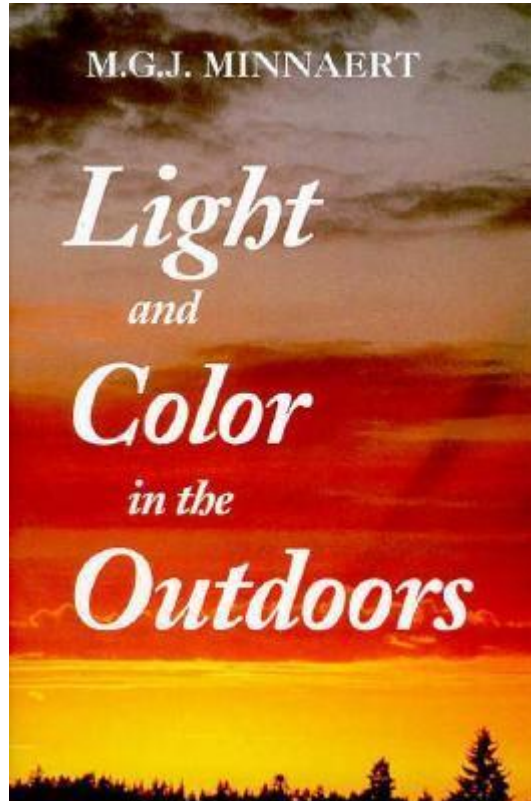


Light size: large



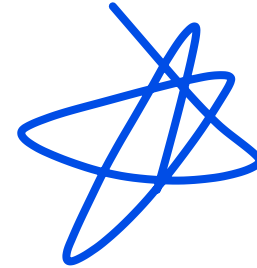
Good book

(Will not help you with passing this course)



Shadows are Complex

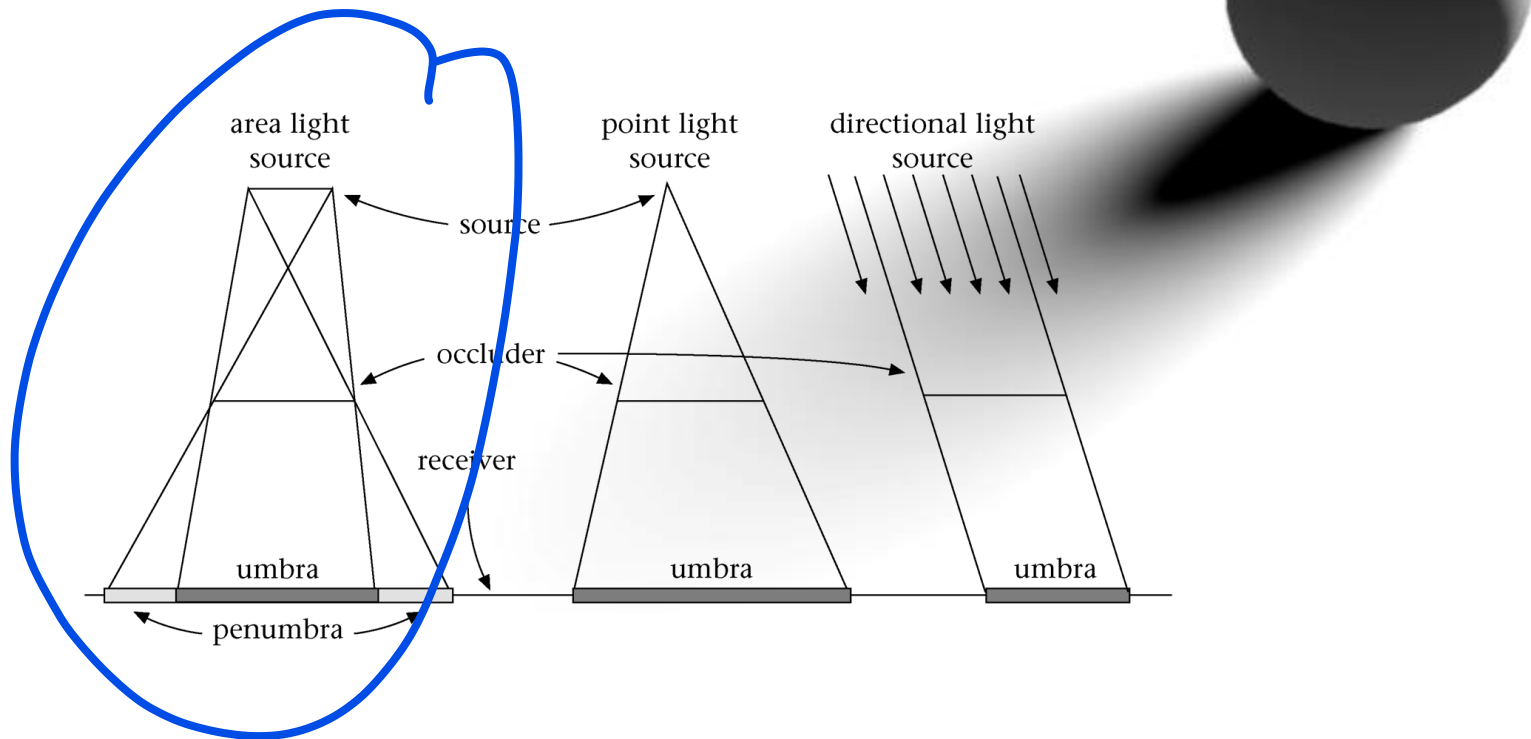
- In the real world sources of light are not points
- The intensity within a shadow is not constant
 - **Umbra**
part that sees nothing of the source
 - **Penumbra**
part that receives some light



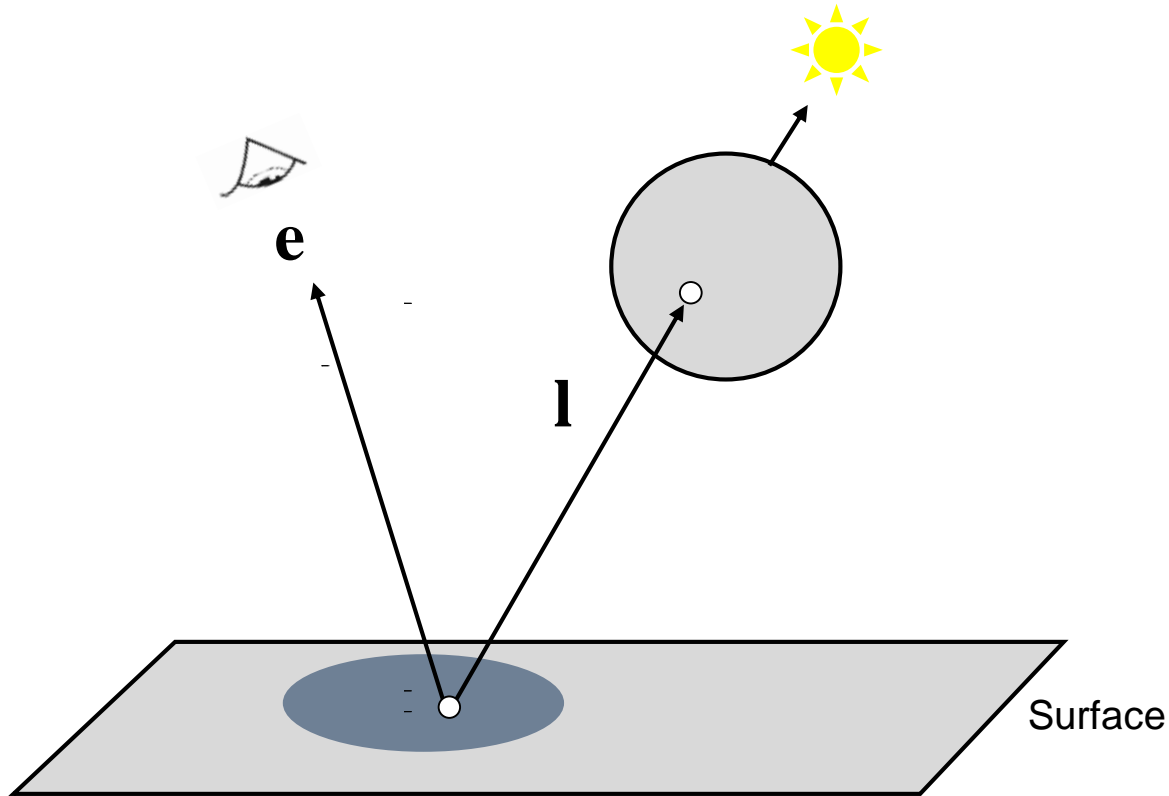
terminology

soft shadows

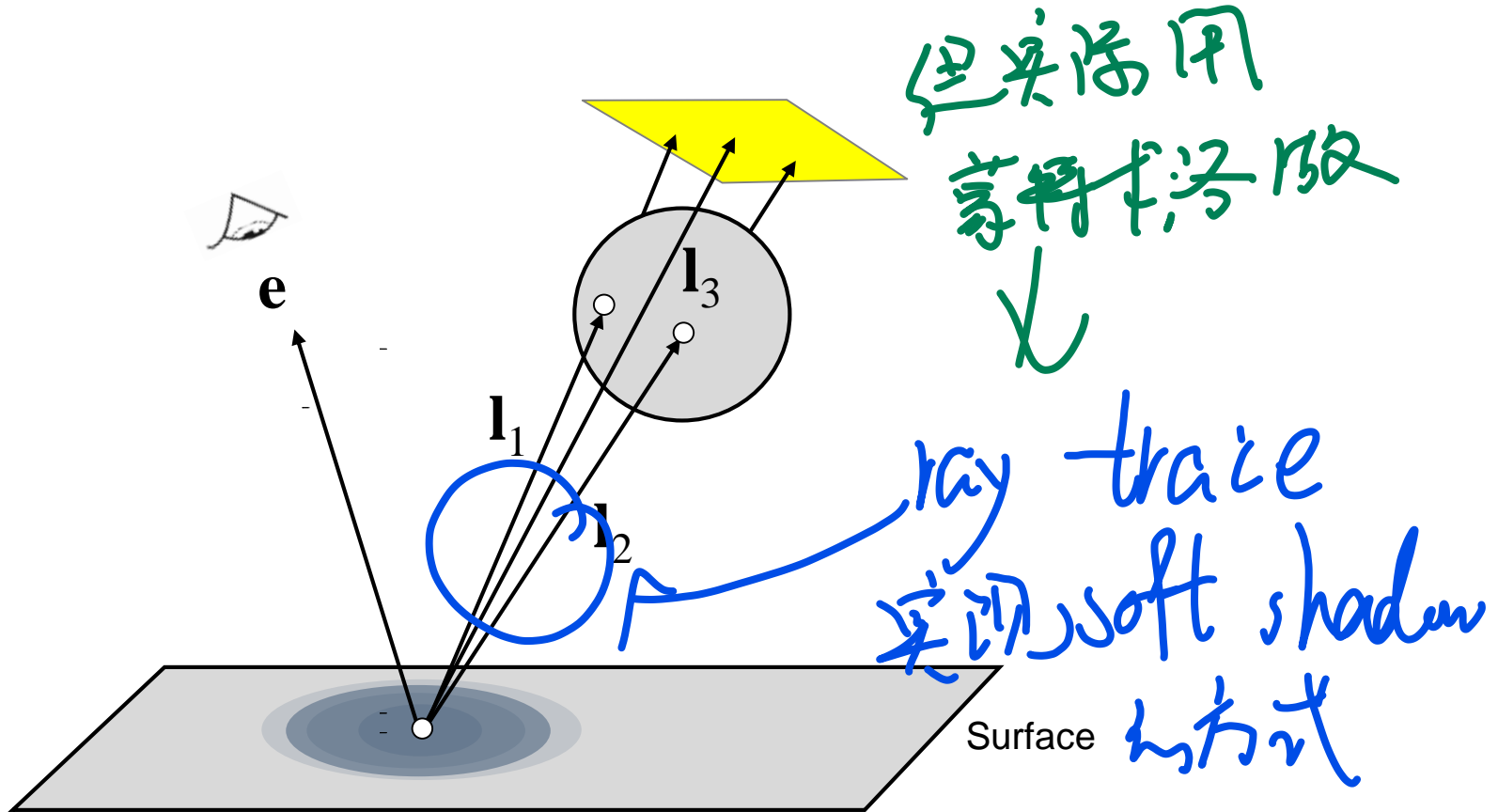
Definition



Shadow in ray-tracing

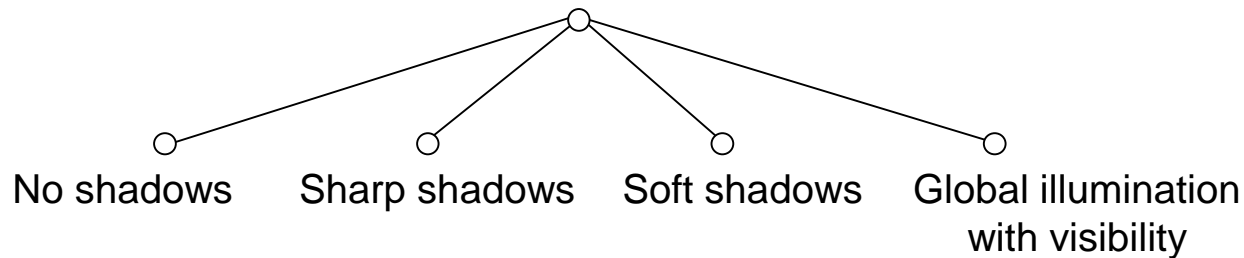


Soft shadows in ray-tracing



Current Shadowing Methods

- There exist a large number of methods
- We are interested in methods suitable for interactive walkthroughs, speed is crucial
- We will classify them by complexity:

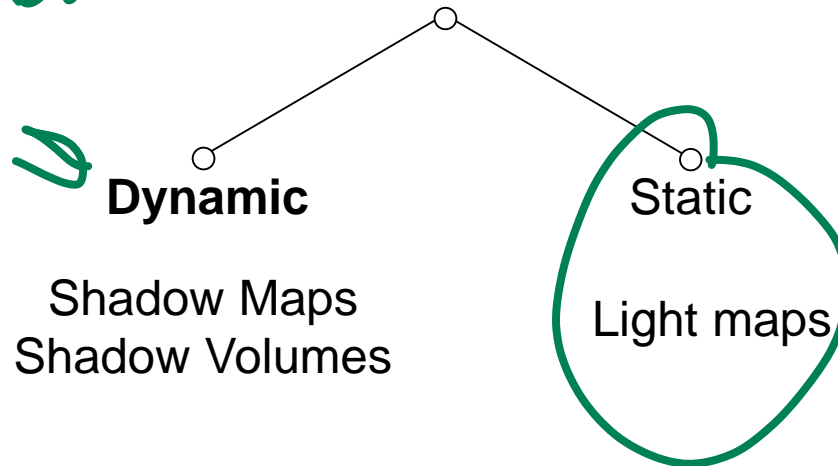


Sharp shadows

Sharp Shadows

- Source is assumed to be a point or direction

lights & models
can move

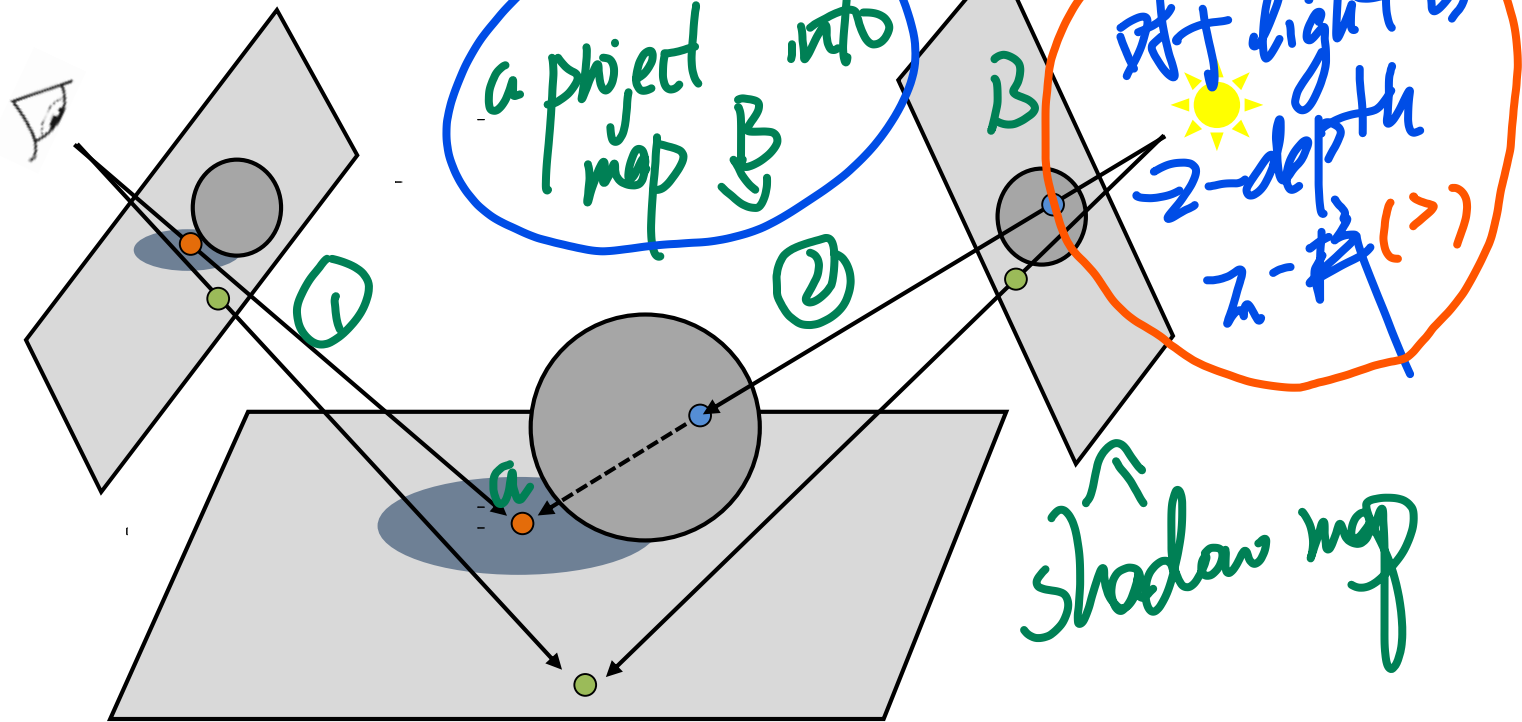


之前学过的

Shadow Maps

Light / shadow duality:

A point is in shadow if it is **not** visible from the light source



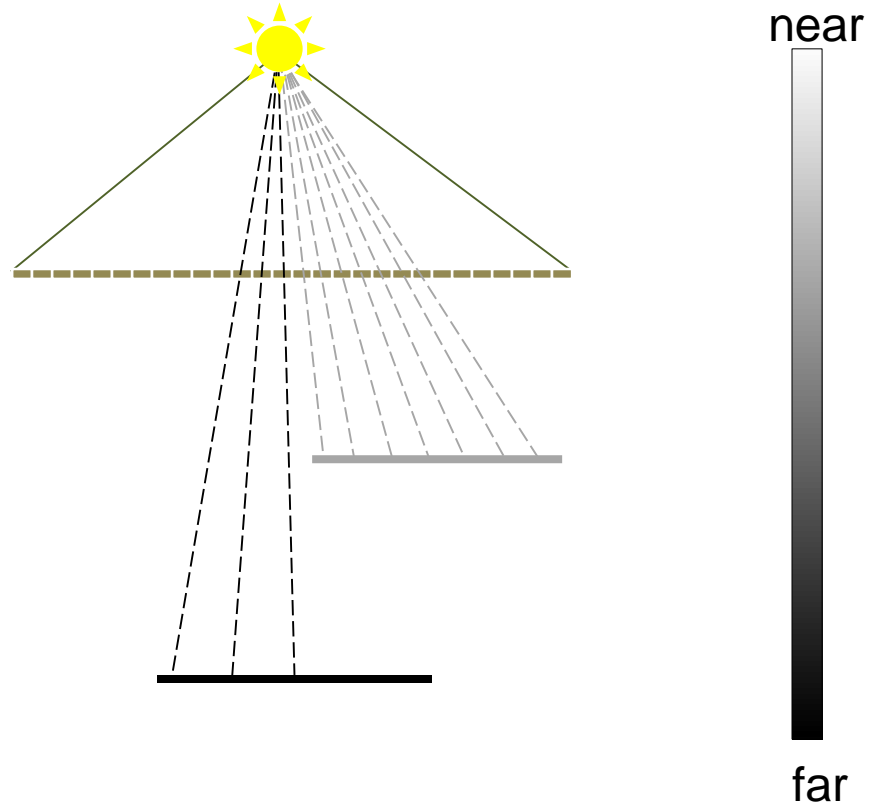
Two passes

- Shadow map pass
 - Use the light source as a view point (light space)
 - Render scene
 - Store depth information in a shadow z-buffer = **shadow map**
- Shading pass
 - Render scene as usual from the camera's view point
 - Each pixel's position (x_v, y_v, z_v) is transformed to light space (x_s, y_s, z_s) ,
 - If
 - the z_s value is less or equal to the shadow map at x_s, y_s it is **lit**,
 - else it is **shadowed**

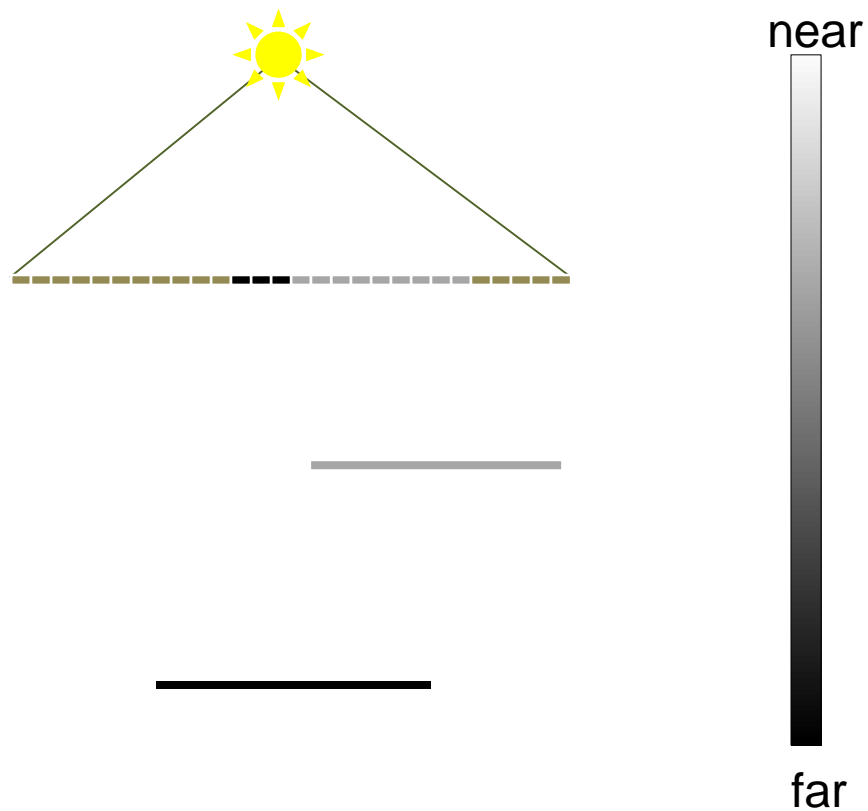
Shadow Maps



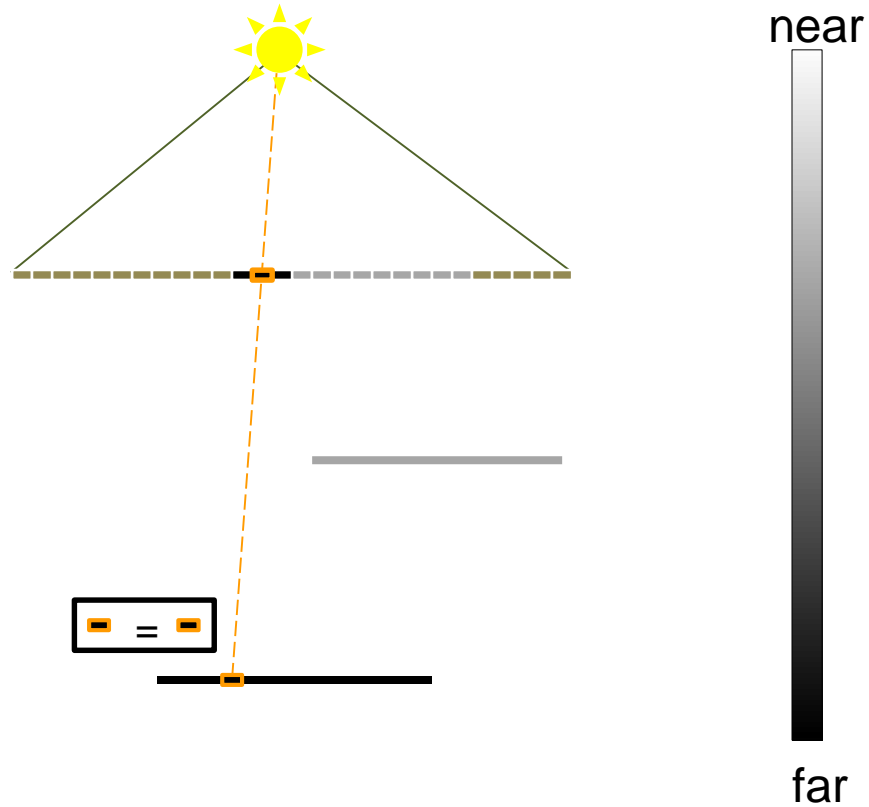
Shadow Maps



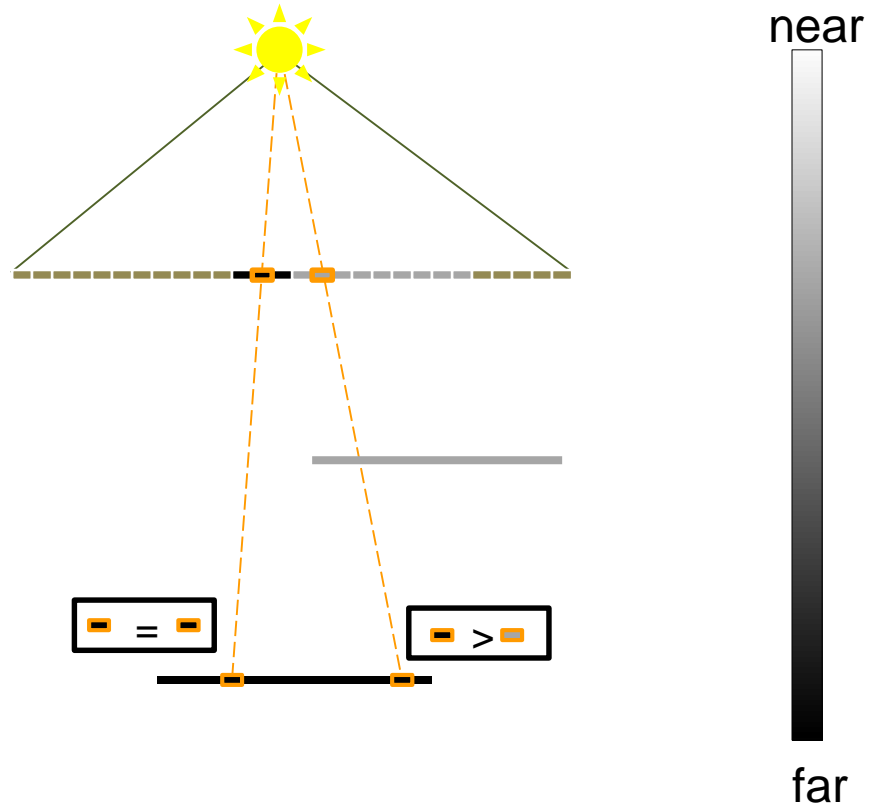
Shadow Maps



Shadow Maps



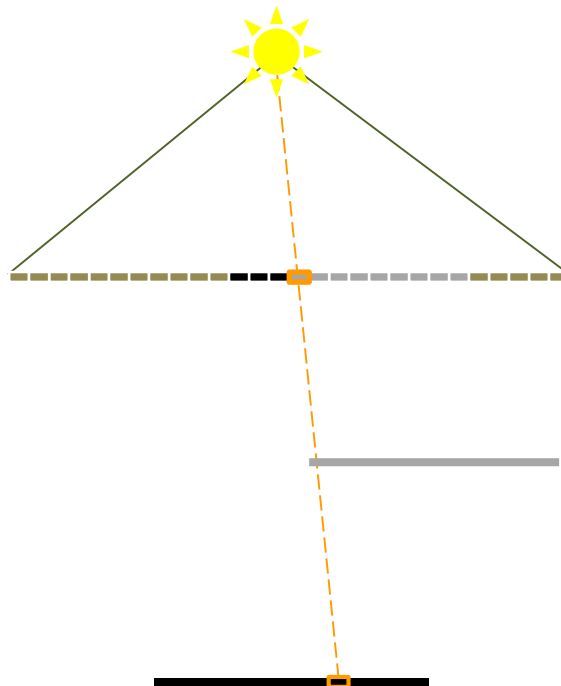
Shadow Maps



Shadow Map Filtering

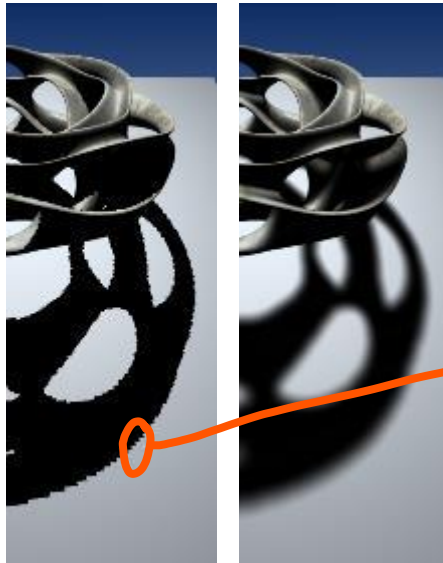


unfiltered



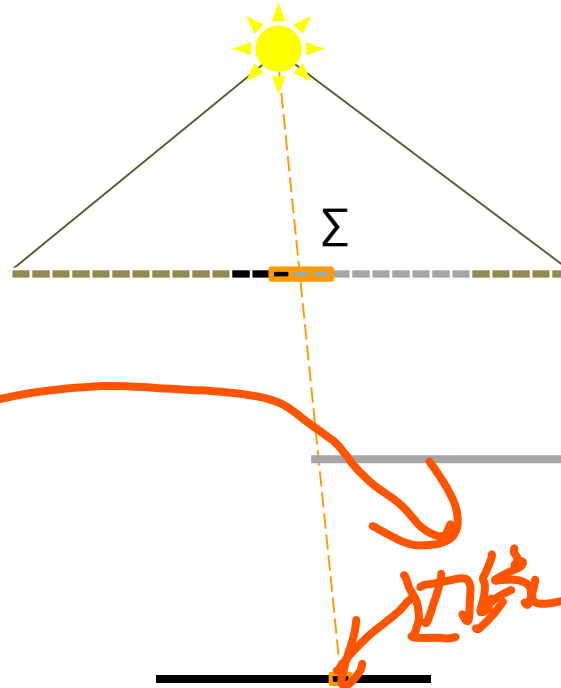
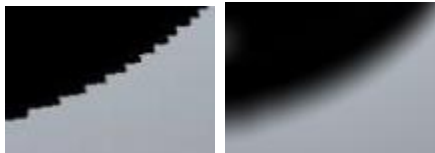
Shadow Map Filtering







多次采样
Shadow Map
LP4F



unfiltered

filtered



| | | |
|---|---|-----|
|  =  | 1 | 33% |
|  >  | 0 | |
|  >  | 0 | |

Percentage Closer Filtering
[Reeves et al. '87]

also take into its
neighbours

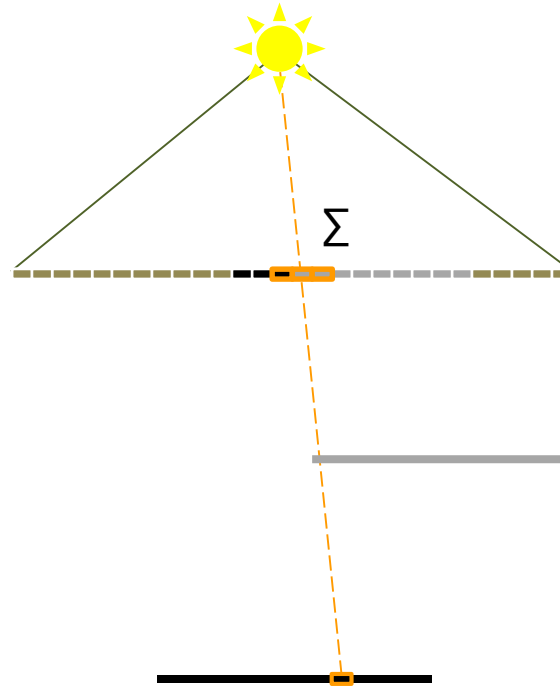
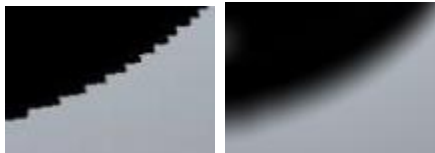
看到邻居遮挡情况

Shadow Map Filtering

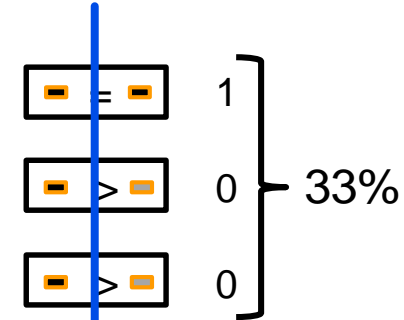


unfiltered

filtered



Percentage Closer Filtering
[Reeves et al. '87]

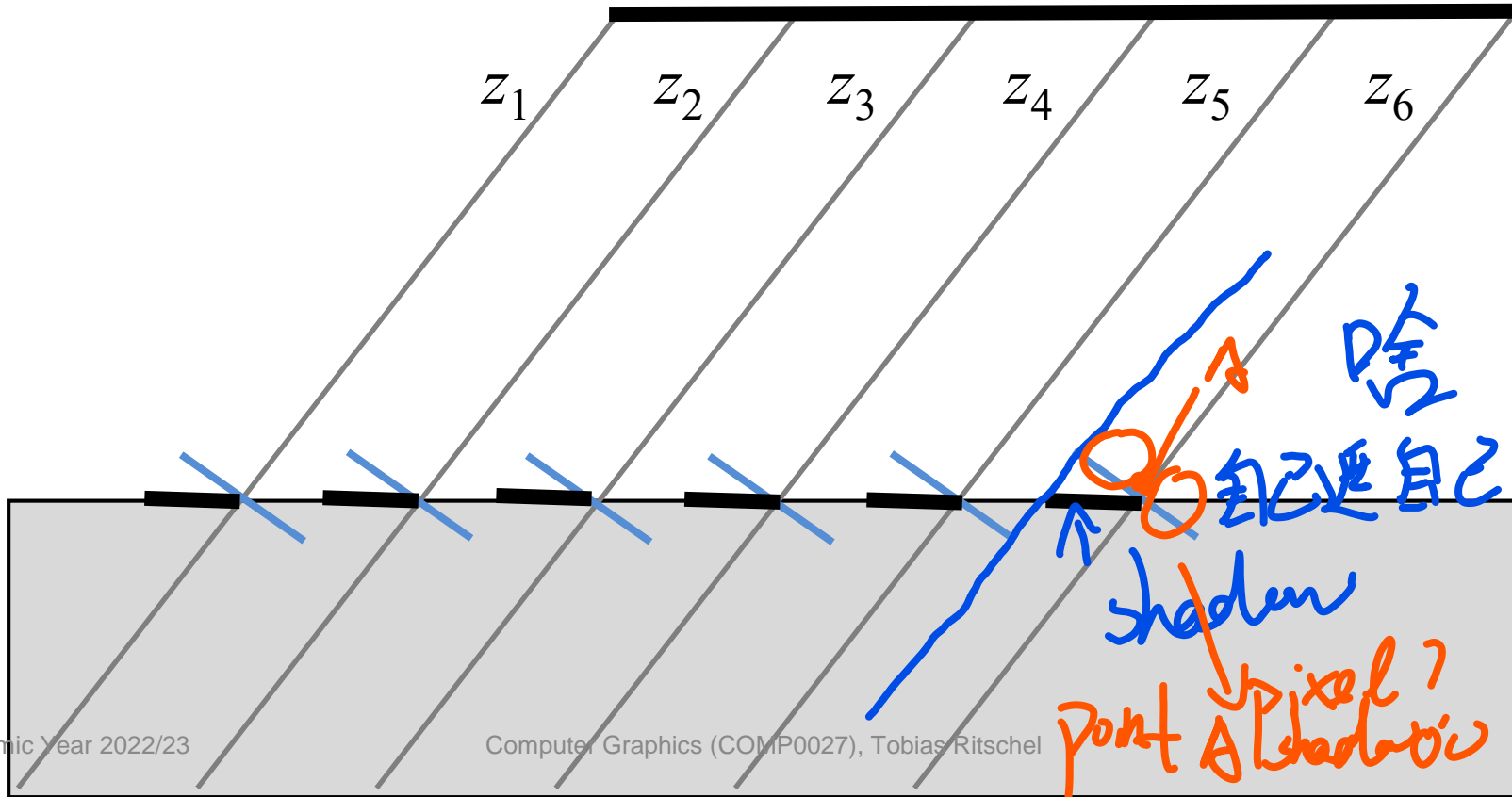
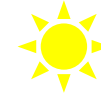


compare first, then filter!

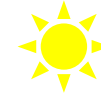
Expensive

这个比较 expensive 还这个 sort py

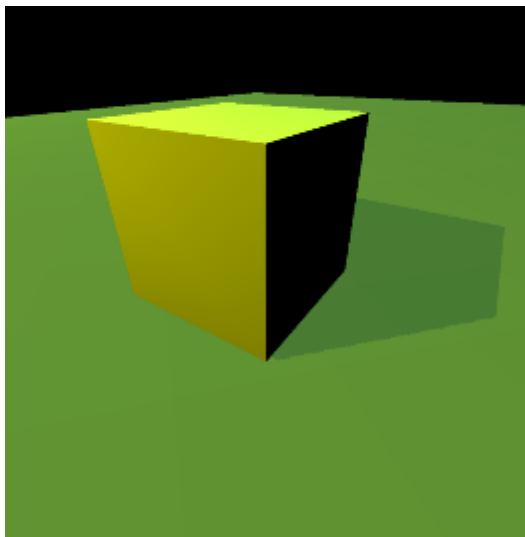
Self shadowing: Disaster



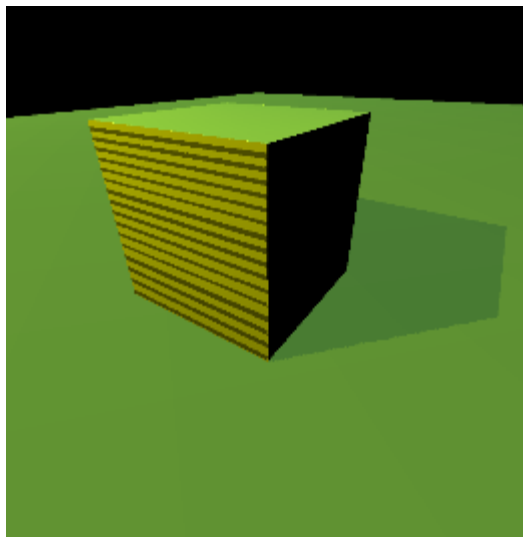
Self shadowing fix

 $z_1 + \epsilon$ $z_2 + \epsilon$ $z_3 + \epsilon$ $z_4 + \epsilon$ $z_5 + \epsilon$ $z_6 + \epsilon$

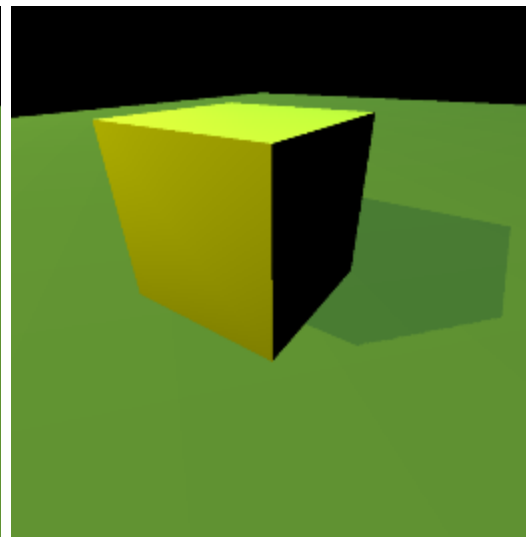
Bias (Epsilon) for Shadow Maps



Correct result



Not enough bias



Too much bias

Shadow Maps

- “Less than or equal” test is imprecise
 - Gives rise to “shadow acne”
- Often found in hardware now
 - Otherwise high-cost operation
- Imprecise since it is only accurate in the image space of the light
 - Imagine a shadow throw over complex objects or long distances
- Quality depends on resolution (jagged edges)
 - Percentage-closer filtering helps
- FOV of shadow map?

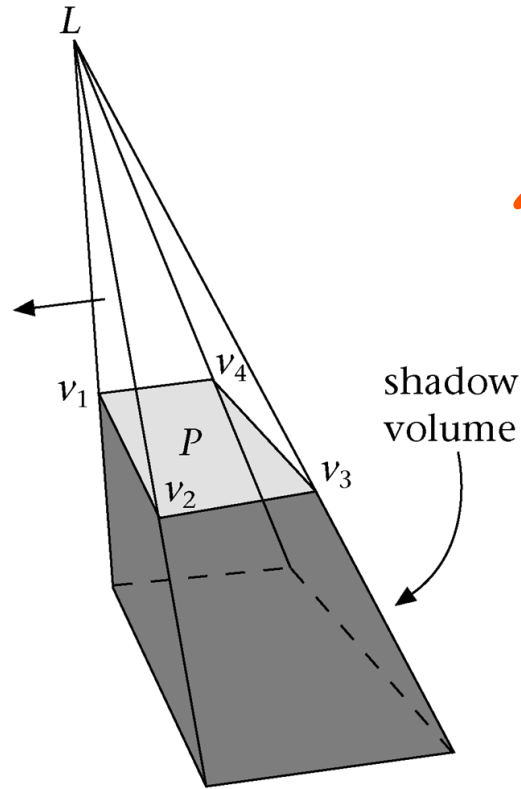
Shadow Volume Method



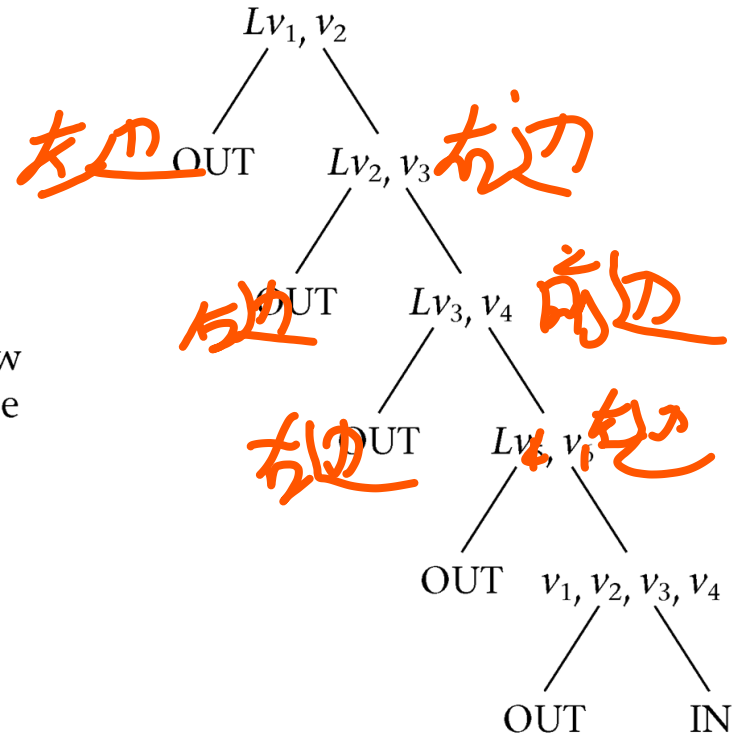
- Shadow volume (SV) is the volume of space below a polygon that cannot see the source (a culled pyramid)
- During rendering of image, the line from a point visible through a pixel to the eye is intersected with all object SVs
- The number of intersections indicates if the point is in shadow or not

check point
whether in shadow volume

Shadow Volumes



(a)



(b)

Shadow Volumes

- Just like a polygon - you are inside a volume if you need to cross a surface to exit it
- General idea of shadow volumes is count the number of shadow planes you cross
 - +1 for front facing
 - -1 for back facing
- If total is >0 you are in shadow
- Special case if the eye itself is in shadow

奇数 (进入次数)

不能这么算



Shadow Volumes

Two stages:

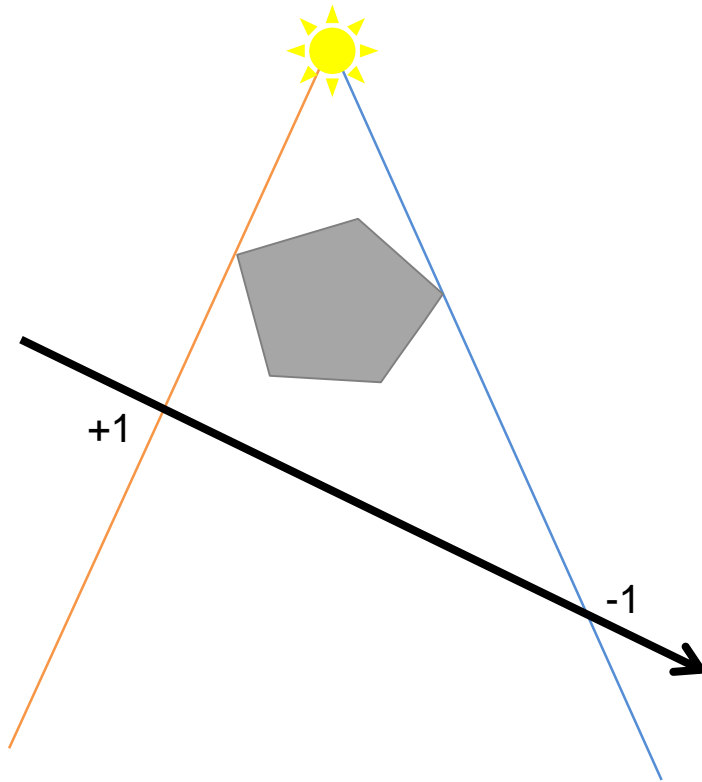
1) Volume **construction**

- Find all planes of the shadow volume and their plane equations

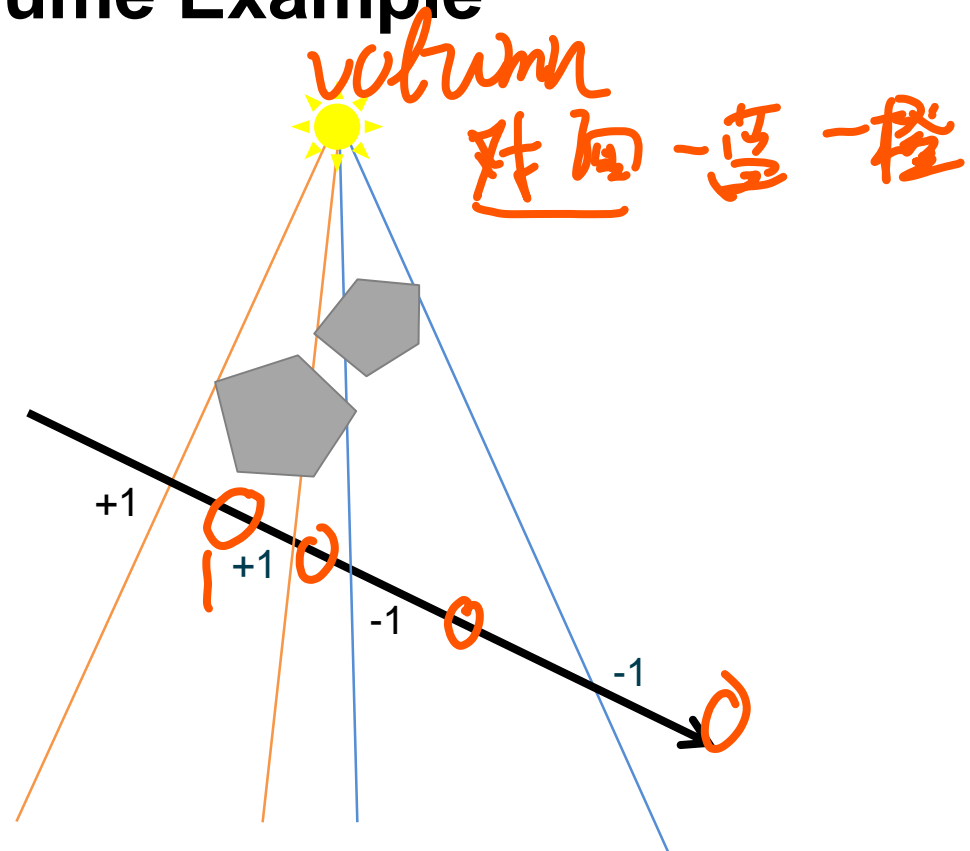
2) Volume **test**

- Determine shadow plane count per pixel
- Use a scan-line method OR stencil test

Shadow Volume Example

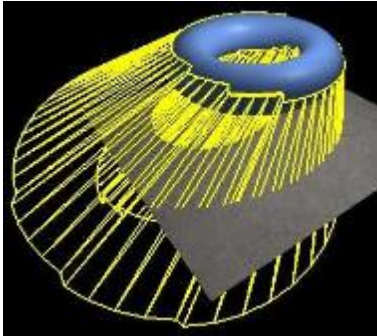


Shadow



Shadow

Shadow Volumes with OpenGL



- Shadow volumes are rendered at each frame
- The stencil buffer is used for counting how many SV are crossed
- Sometimes not all objects are used for casting shadows


Shadow Volumes & Stencil Test

模板测试

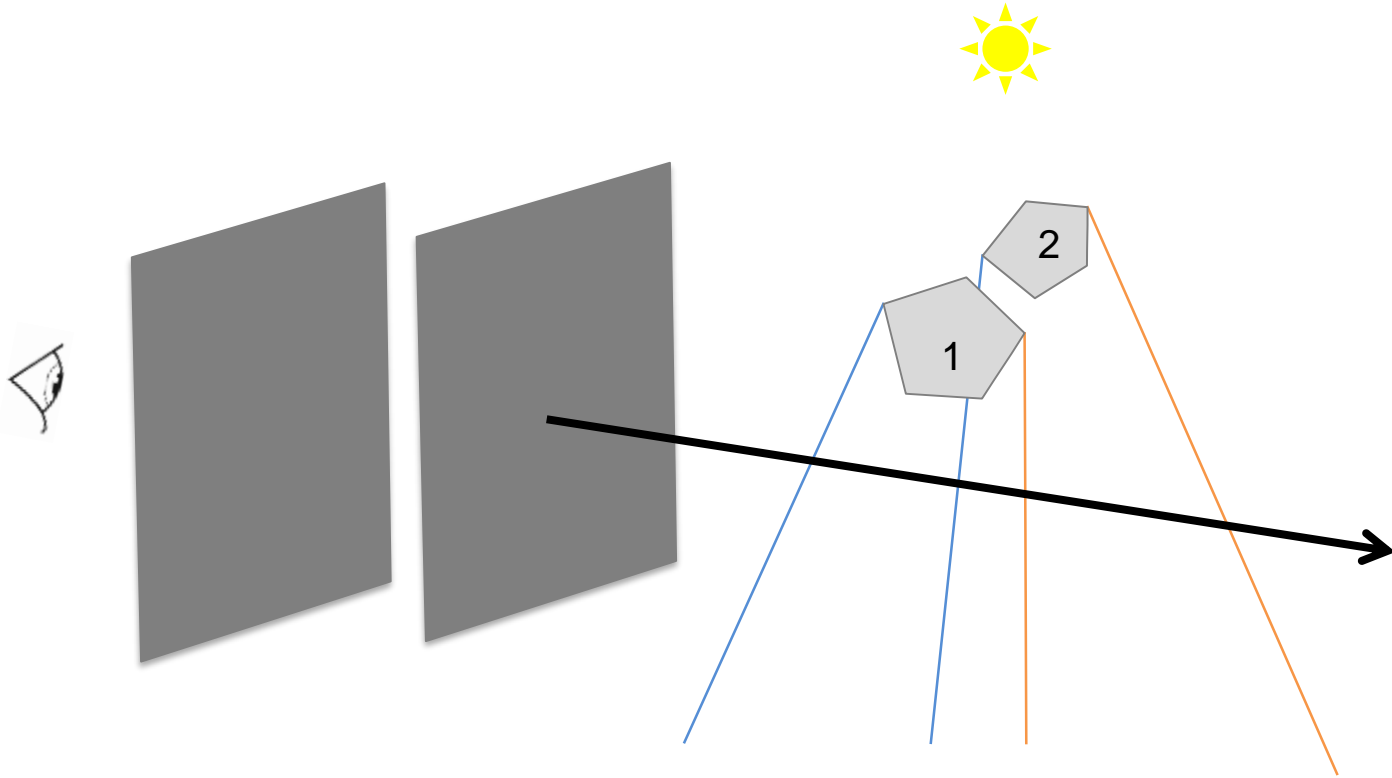
- A stencil buffer is screen sized buffer (1-8 bit) that stores a flag about a rendering operation
 - E.g., `stencil[x, y]` is negated if `zbuffer[x, y]` is less than current `z` value (i.e. stencil is set if and only if `z` buffer-test passes)
- Many uses in graphics

Stencil Test 是用
bit - stencil Test 是用
shadow map 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

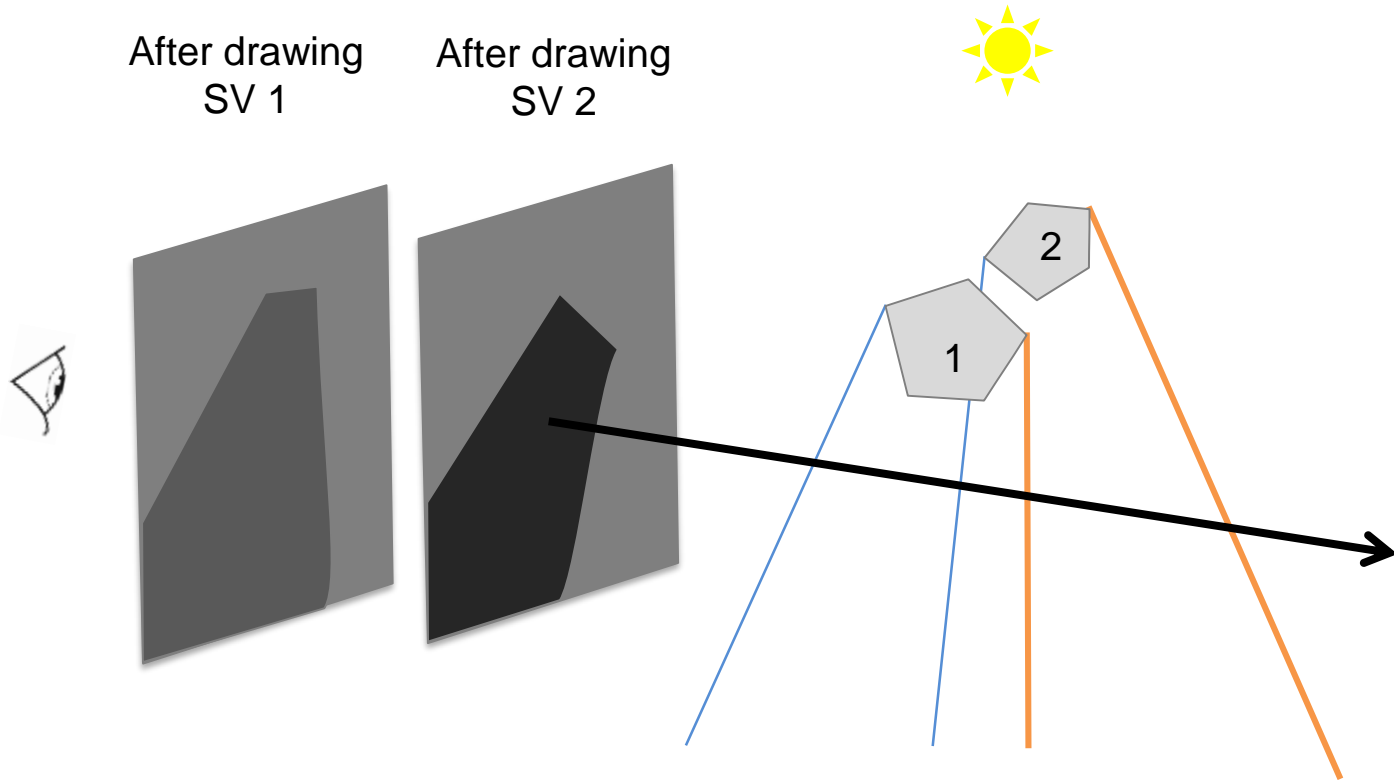
Shadow Volumes & Stencil Test

- 
- Render the scene into the RGB and z -buffer
 - Turn z -buffer **writing** off
 - Render all shadow volume polygons with the stencil buffer
 - Increment stencil count for front-facing
 - Decrement for back facing
 - Re-render scene with lighting **off** and only render pixels where stencil is non-zero

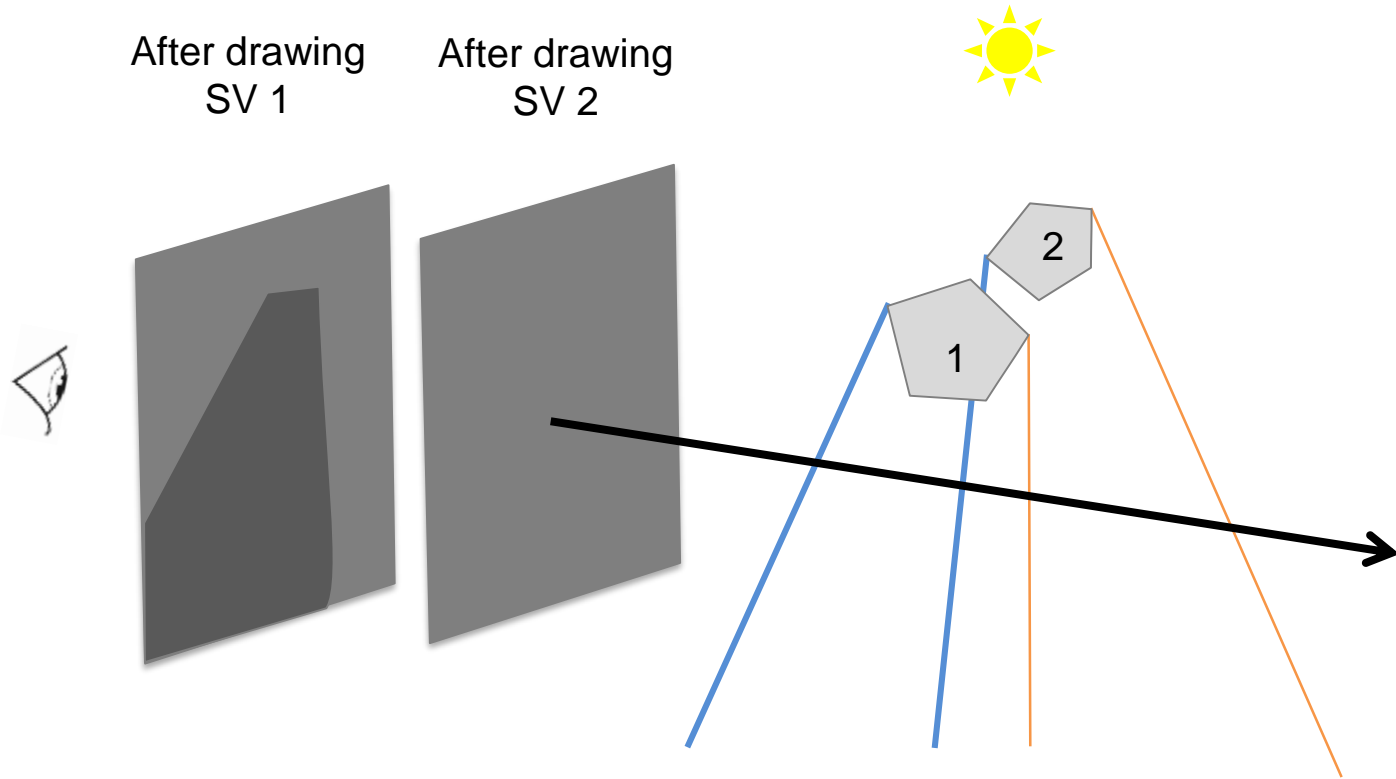
No-shadow example



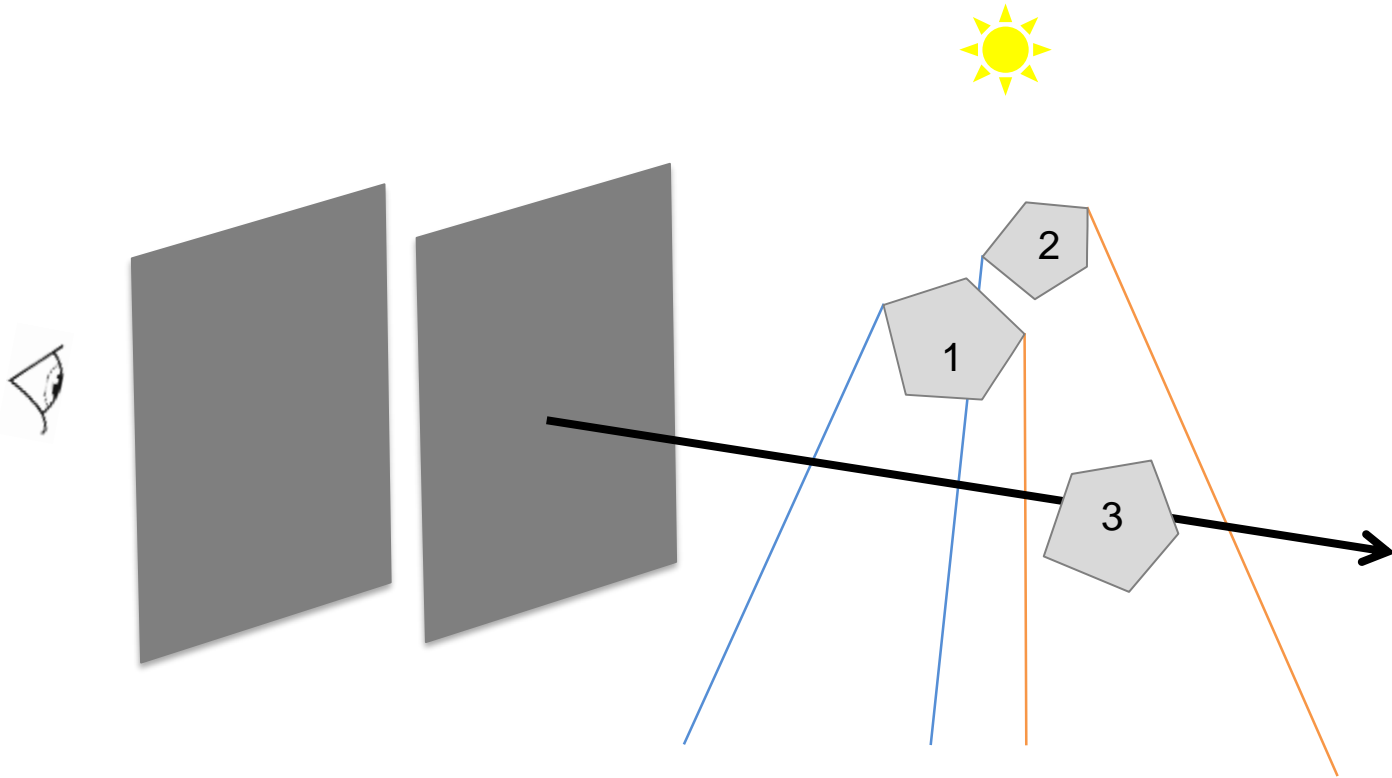
No-shadow example



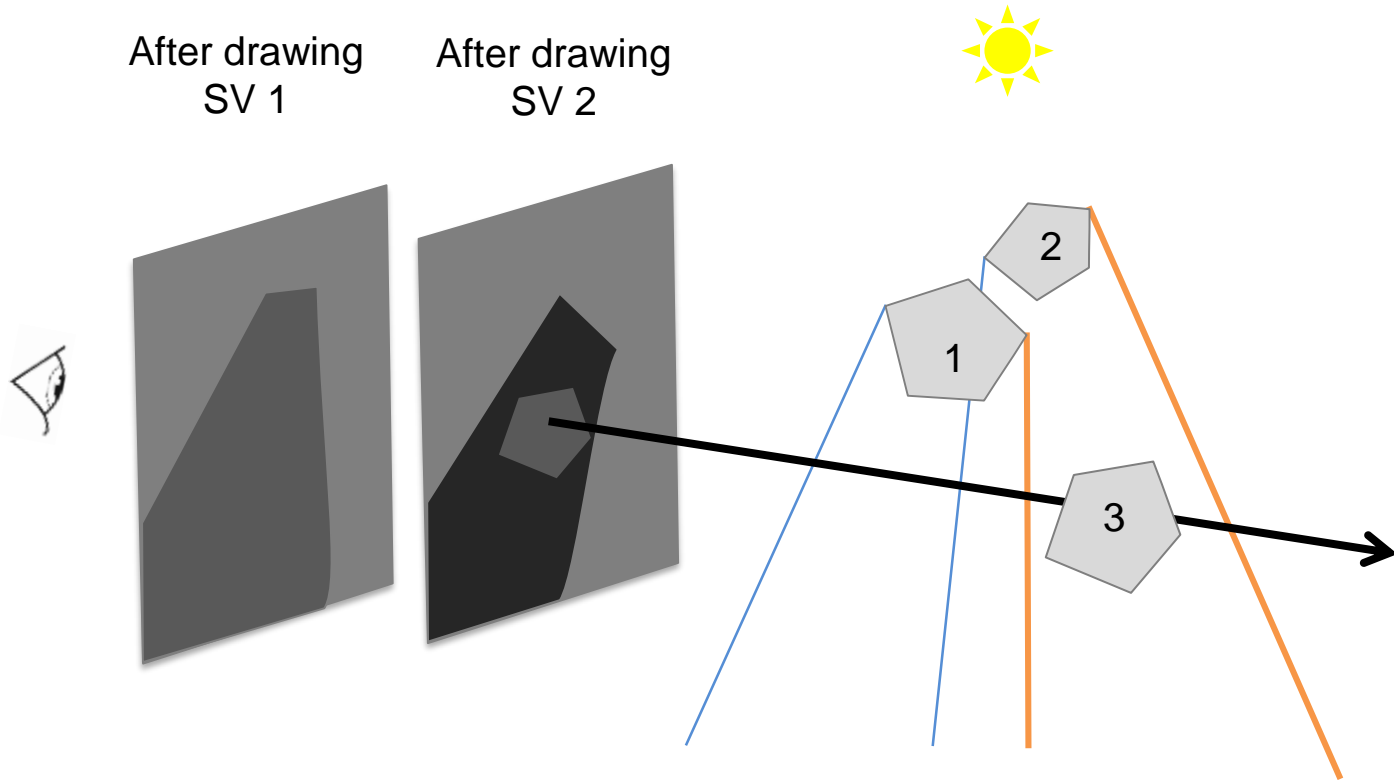
No-shadow example



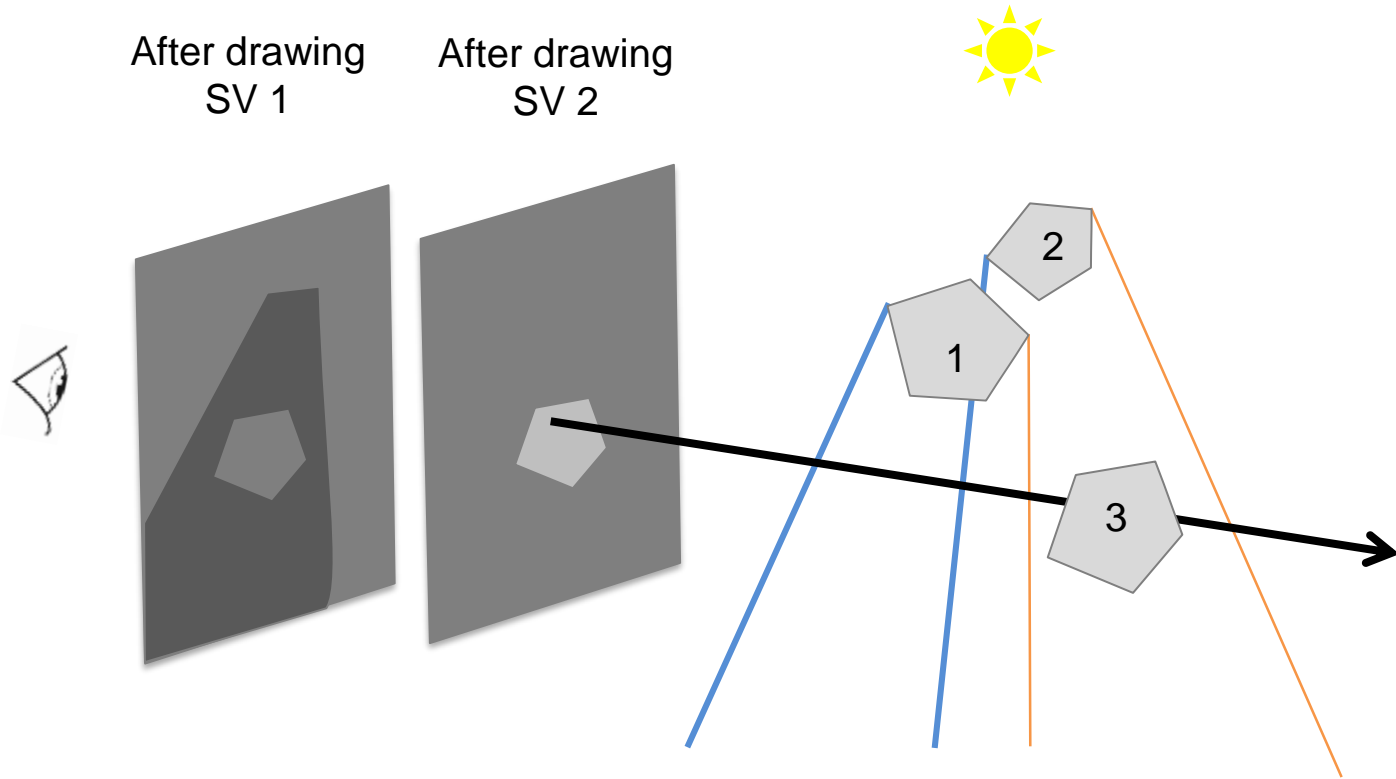
Has-shadow example



Has-shadow example



Has-shadow example



Summary for Sharp Shadows

- Four shadow umbra techniques
- Image space
 - Shadow maps
 - Shadow volumes

Soft Shadows

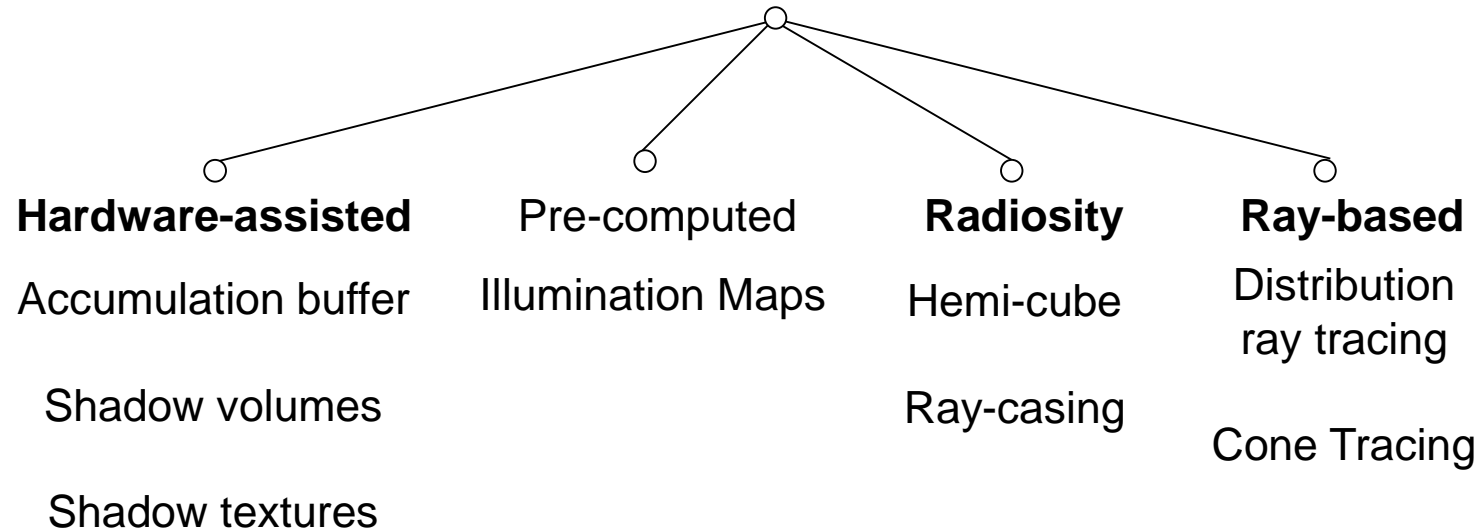
↑
他没讲过
回去看

Soft Shadows

- Source has a finite extend
- Images look a lot more realistic



Soft Shadows



Analytical v. Sampling

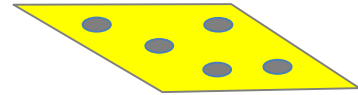
- Analytical
 - Find all boundaries within the penumbra. Done almost exclusively for polygonal light sources
- Sampling
 - Approximate solution that treat the light source as a set of points. Any shape source is possible.

↑
上面是 penumbra
一半影

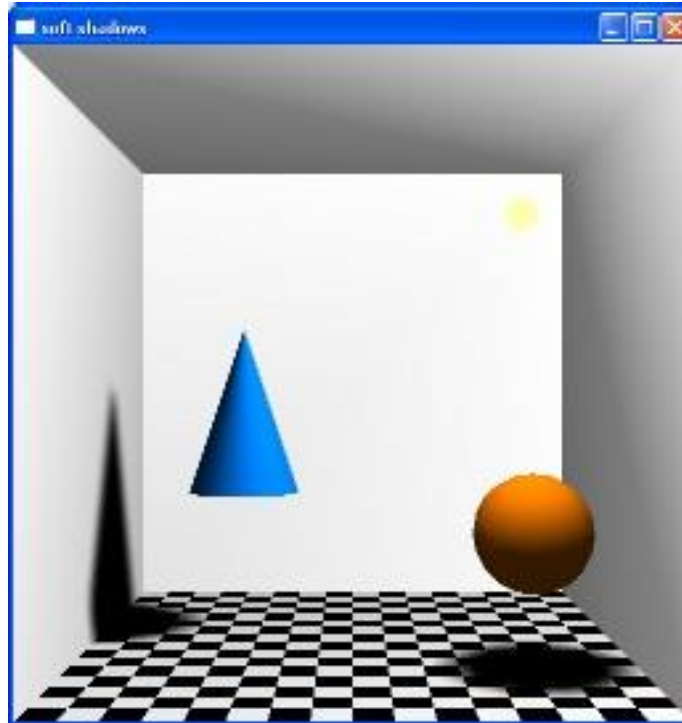
↑
在一个 shadowmap 中
用号做快
为一个不在 shadow
里

Soft Shadows using Point Sampling

- Place many point lights on an area light
 - Random positions work just fine
- Render hard shadows from each point light
 - E.g., using shadow volumes or shadow maps
- Sum up all contributions
 - Can be done on the GPU (in the frame-buffer)
- Similar to what ray-tracing does to get soft shadows



Example



Illumination Maps (Shadow Textures)

- Shadows are pre-computed and stored as textures on the receiving polygons
- Displayed using graphics hardware in real-time
- Often use: Radiosity / Path tracing / Photon mapping
- Sometimes called “baked” lighting, very common in game engines
- Disadvantage: lighting cannot change

Recap

- In order to regain shadows in real-time engines, we have to do a lot of work
- A very large number of shadow algorithms exist
- Many of them are unsuitable for walkthroughs of very complex scenes:
 - with pre-computation methods scene cannot be modified
 - or are too slow (ray-tracing, soft shadows)
- Hard shadows
 - on-the-fly methods (SM and SV) are fast enough