CS 5 finale

Looking back! Evals, Ideas

Looking ahead? Options…

I'll be back…

That's my line!

CS 5, on the verge of *termination*

CS 5 Final Projects

due this Fri. 5:00 pm (or later w/ Euro)

Friday hours at the LAC !!!

CS 5 Review Session

7-8+ pm Sunday, 12/15 in big Shanahan

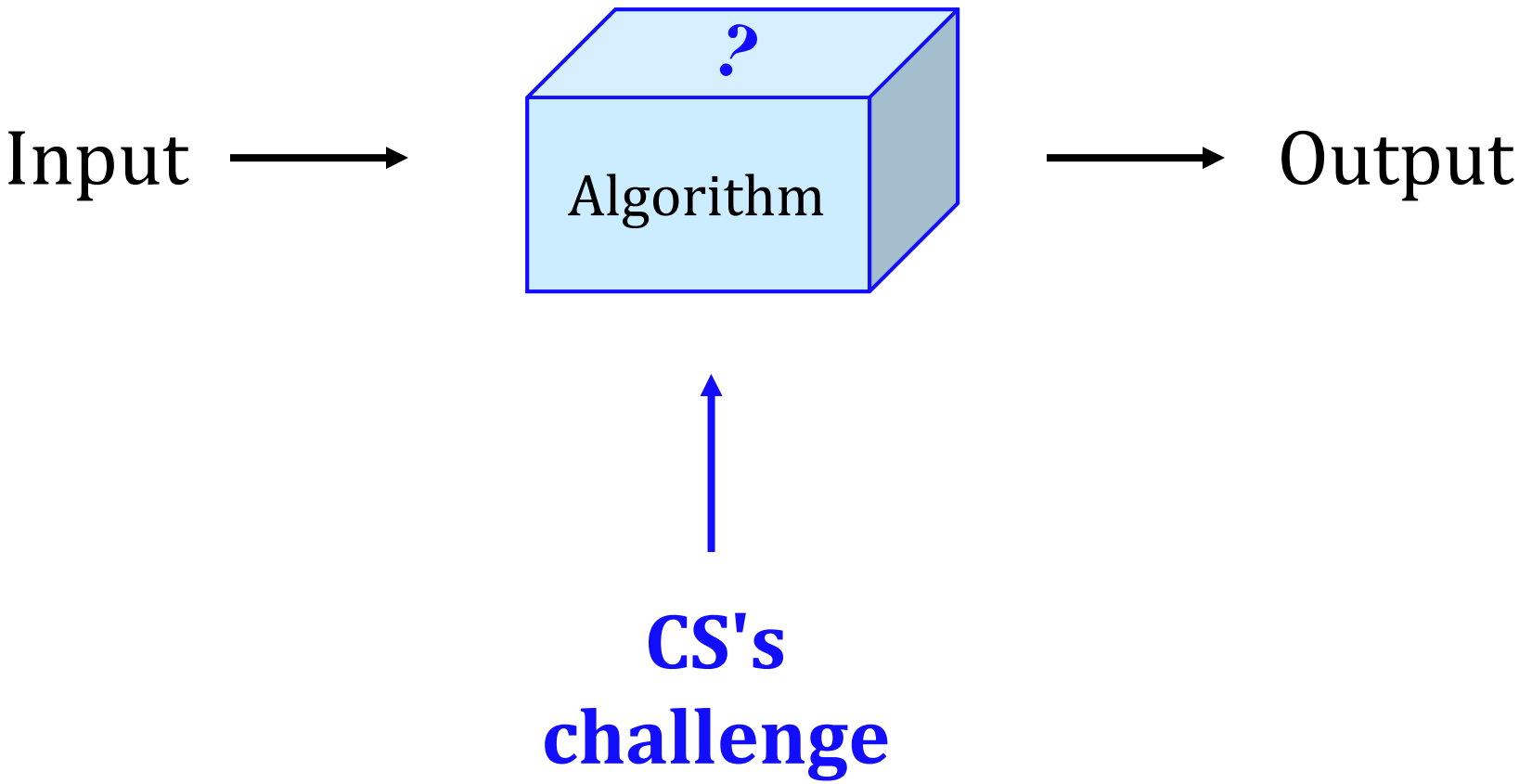
CS 5 Final Exam

2pm Mon. 12/16 or 2pm Wed. 12/18, in big Shan

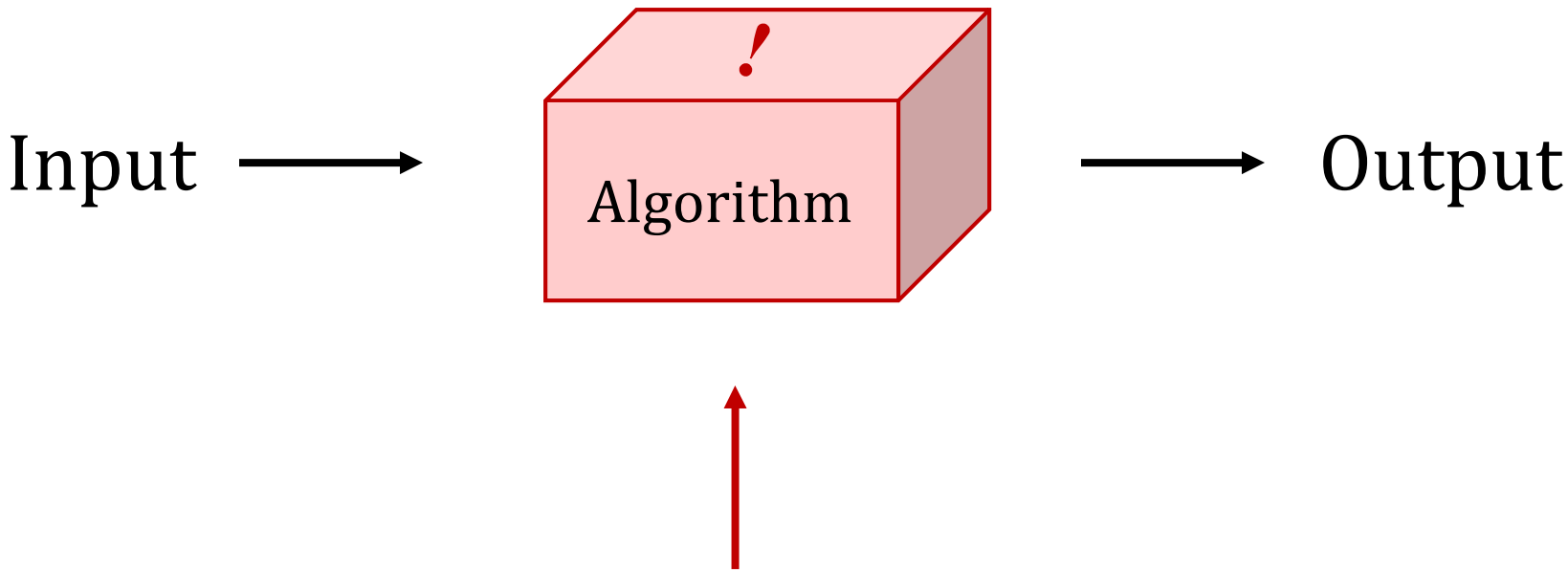
Two pages of notes are OK

practice probs, hwks, quizzes to review

The CS view of the world...



Uncomputable functions



Sometimes an algorithm simply does not exist…

more precisely: every possible algorithm contains bugs!

Haltchecking is uncomputable.

Halt Checking

`hc( f )`

~ returns whether **`f()`** halts or not

`hc` can't be written

well, not without bugs

Corporate Address

Apple
1 Infinite Loop
Cupertino, CA 95014

Halting problem

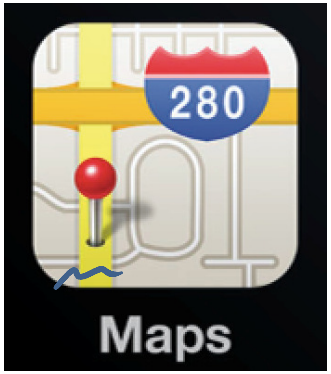
From Wikipedia, the free encyclopedia

In computability theory, the **halting problem** can be stated as follows: Given a description of a computer program, decide whether the program finishes running or continues to run forever. This is equivalent to the problem of deciding, given a program and an input, whether the program will eventually halt when run with that input, or will run forever.

Alan Turing proved in 1936 that a general algorithm to solve the halting problem for *all* possible program-input pairs cannot exist. A key part of the proof was a mathematical definition of a computer and program, what became known as a Turing machine; the

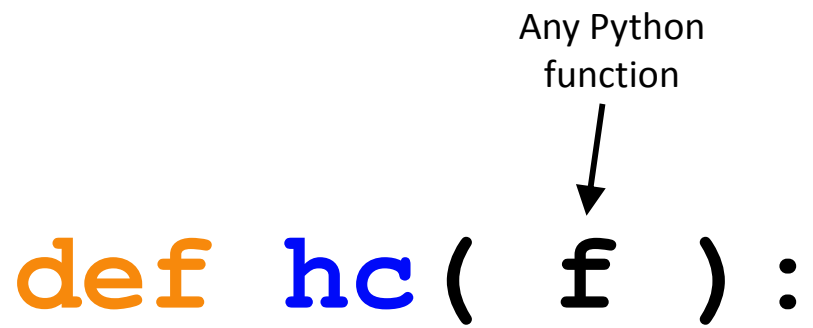
Corporate Address

Apple
1 Infinite Loop
Cupertino, CA 95014
408.996.1010



the iPhone's icon for Google Maps …

Haltchecking is uncomputable.



It is <u>impossible</u> to write a bug-free function **`hc(f)`**
that determines if function **`f`** halts or loops forever:

(1) **`hc(f)`** returns **`True`** if **`f()`** halts and
(2) **`hc(f)`** returns **`False`** if **`f()`** loops infinitely

Haltchecking a few functions...

```
def f():
  while True:
    print 'Ha!'
```

```
def g():
  return 42
```

```
def BFF():
  if hc(BFF) == True:
    while True:
      print 'Ha!'
  else:
    return 42
```


`hc(f)` **`hc(g)`** **`hc(BFF)`**

Should `hc` return `True` or `False` in each of these cases?

The ultimate T/F quiz!

Haltchecking is uncomputable.

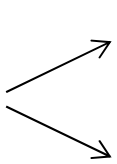
Proof!

Suppose `hc(f)` works for all `f`...

Then write this f'n, `BFF`:

```
def BFF():
  if hc(BFF) == True:
    while True: print 'Ha!'
  else:
    return 42
```

What is `hc(BFF)`



True?

False?

Bug!

Why are these uncomputable?

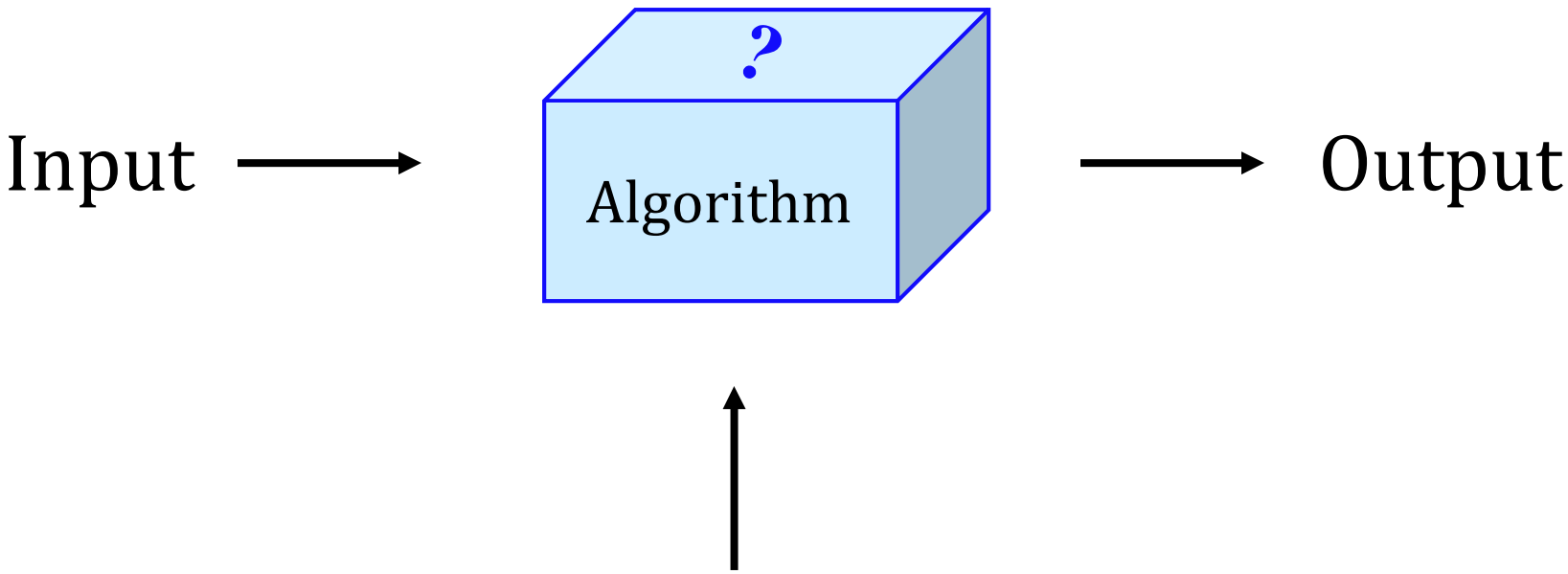
Where did `kc` and `hc` go wrong?

They involved computing what *other programs* were going to do

useful – but impossible!

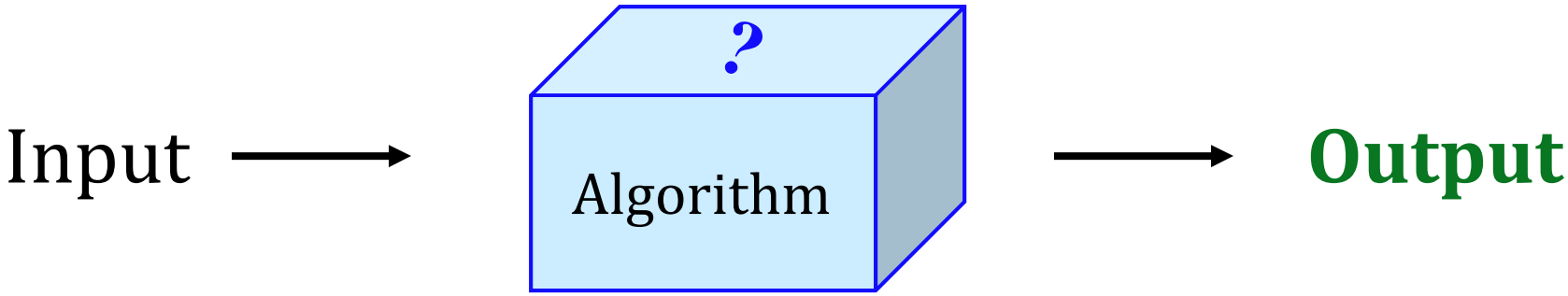
Rice's Theorem: *programs* ***<u>can not</u>*** *determine the results of other programs.* If they try, they always have bugs.

Meaningful functions?



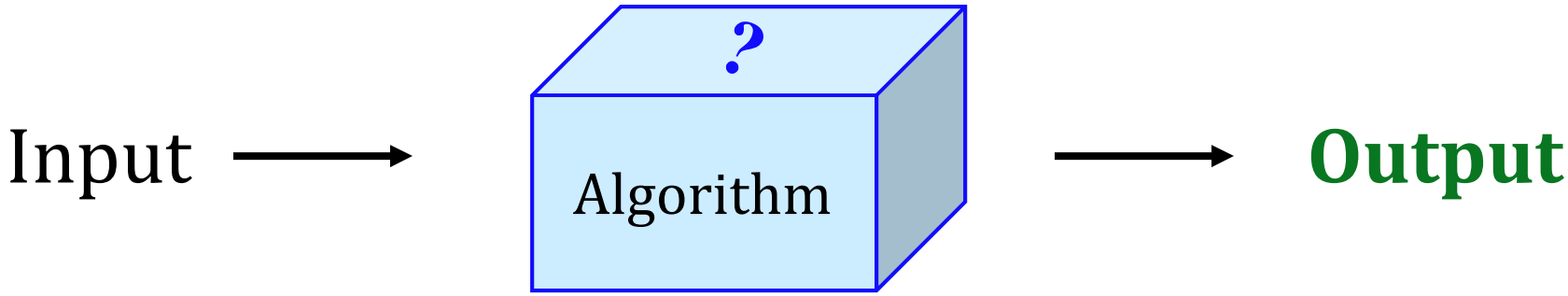
but nearly all ***meaningful*** functions *are* computable…

CS 5's examples…



Connect 4 Board	→	**intelligent move**
Input text	→	**Markov model + new text**
current gener. of "life" cells	→	**next gener. of "life" cells**

CS 5's examples…



Connect 4 Board

Inpu

current
of "life"

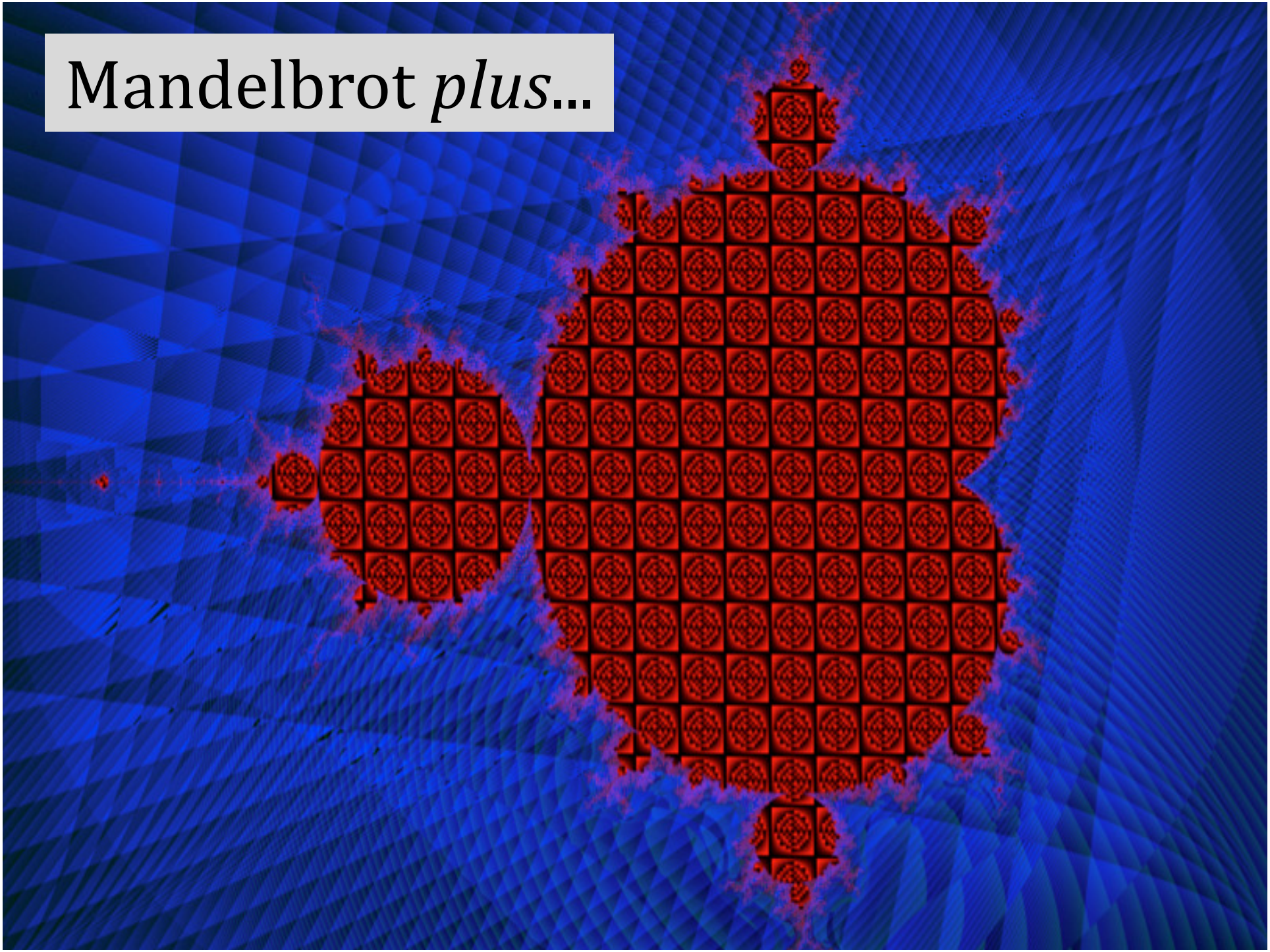
balance of utility & creativity

lligent
ove

model
new text

next gener.
of "life" cells

Mandelbrot *plus*...



Connect Four results…

"Science without religion is lame, religion without science is blind."
"Two things are infinite: the universe and human stupidity; and I'm not sure about the universe."
"Duct tape is like the force, it has a light side, a dark side, and it holds the world together"
"If you die in an elevator, be sure to push the Up button."
"All generalizations are false, including this one."
"Clearly you've never been to Singapore!"
"Luke, I am your father."
"To be, or not to be."
"You shall not pass!"
(… and many others …)

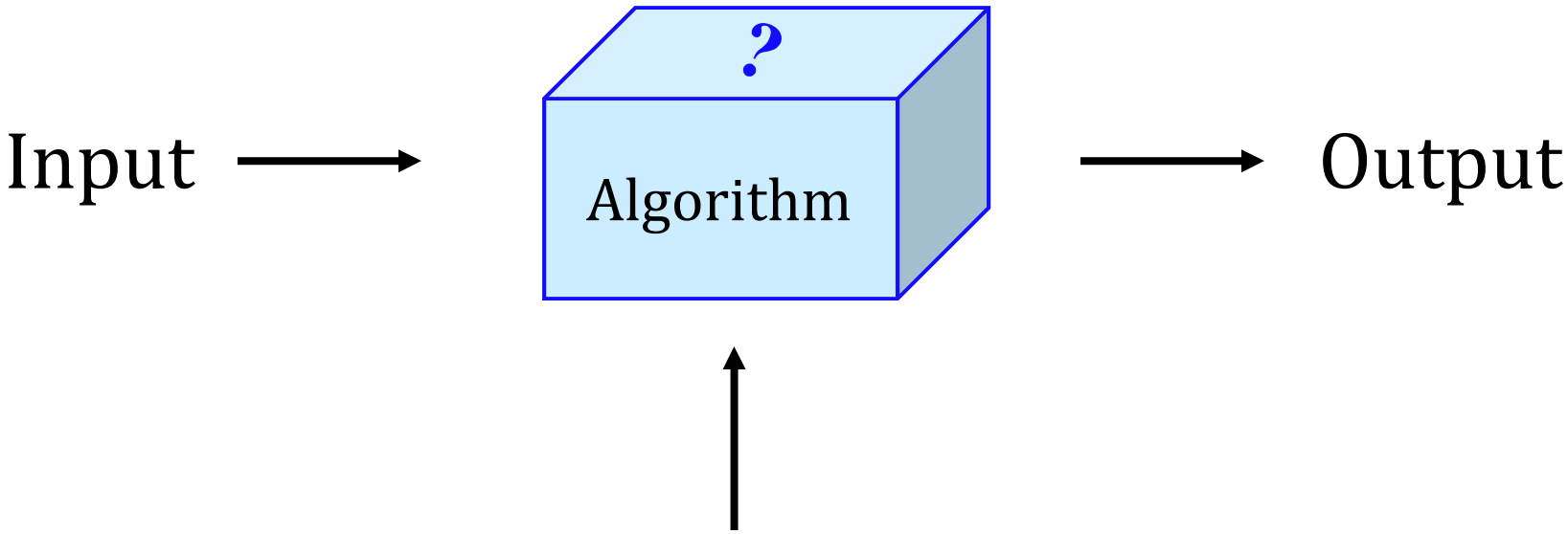
INPUTS

OUTPUTS

"I have a dream! Duct tape is written on. Luke, I am your thoughts and what lies within us."
---- Audrey Rooney

"Your work is lame, religion is lame, religion is nearly the Up button."
---- Abraham Marx

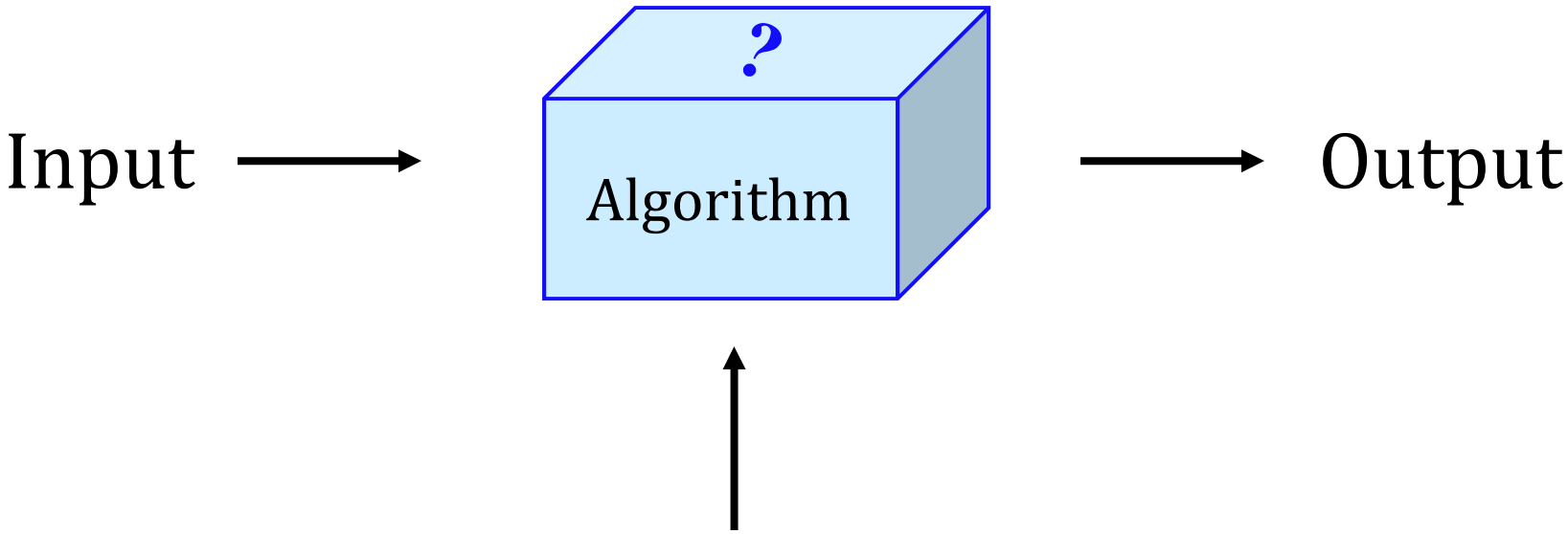
"Two things are false, including this one."
---- Captain_Jack Truman

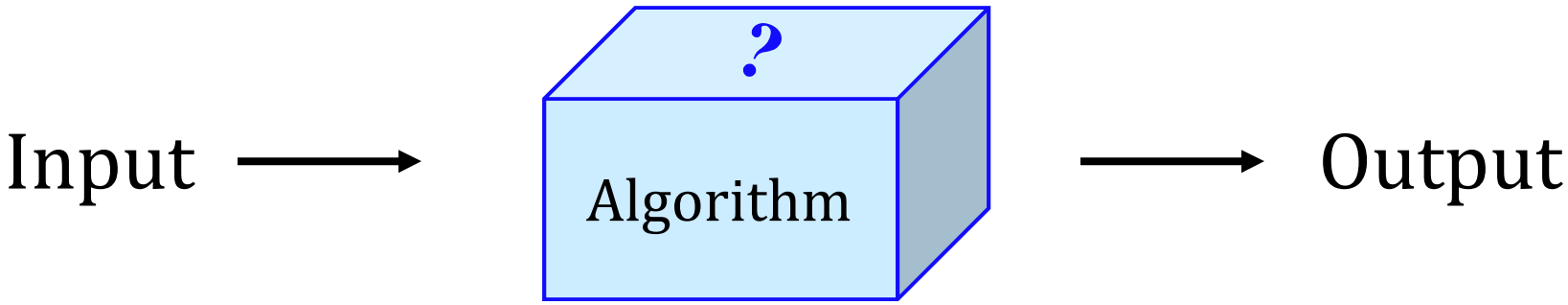


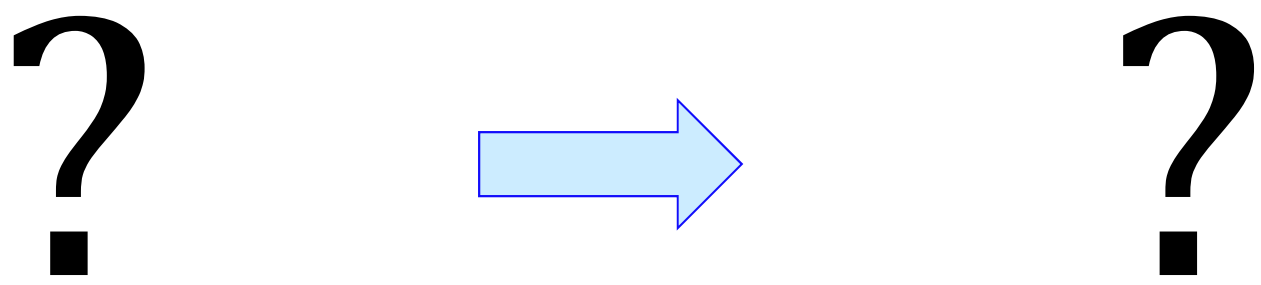
nearly all ***meaningful***
functions *are* computable…

but this doesn't mean we know how to solve them (yet)!

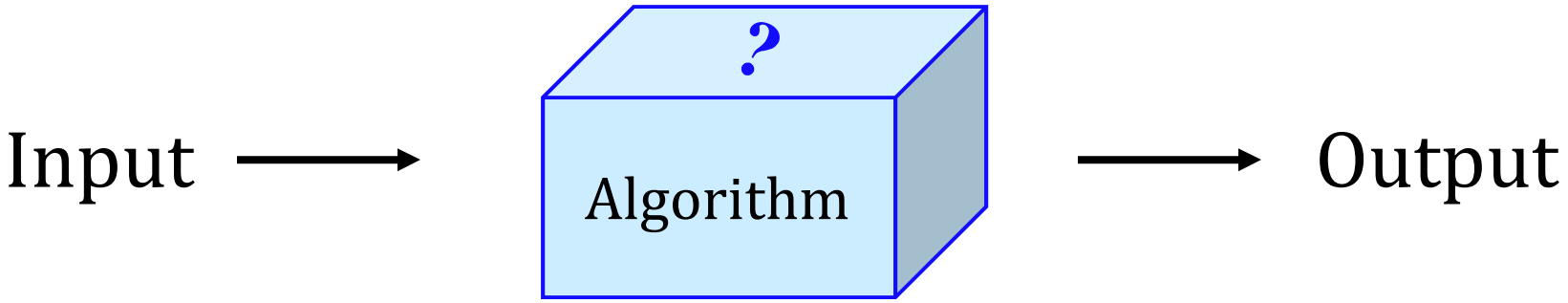
for example, …


Input
?
Algorithm
Output
the "computer vision" problem

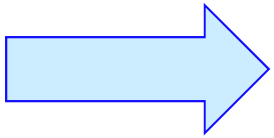




computer vision: *what's* the *input* and *output*?



020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
047 109 107 118 107 115 110 120 120 124 120 128 124 132 131
047 125 130 130 122 121 117 142 131 133 134 141 149 144 135
051 139 143 139 147 134 149 069 127 144 139 144 150 161 149
054 136 161 148 147 158 055 052 034 030 158 156 165 163 156
043 144 165 159 154 171 224 191 047 030 171 165 175 164 163
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173
011 091 101 105 177 039 078 060 041 026 073 102 167 208 121
011 091 094 066 094 033 199 184 139 024 060 094 125 152 134
009 068 072 072 065 031 151 171 075 028 035 072 083 109 063
013 068 074 059 057 037 161 129 062 028 035 071 072 078 056
012 042 063 055 072 033 020 067 031 022 027 082 070 073 060
011 037 064 094 091 026 025 080 066 026 023 071 070 080 060
011 060 077 082 037 023 024 147 140 038 023 037 043 076 037
013 049 076 059 032 028 174 197 182 060 021 021 121 101 062
013 059 111 072 020 078 200 211 182 061 069 059 043 086 106
007 053 057 092 023 105 189 230 210 084 034 021 017 033 091
011 061 072 018 027 054 069 068 062 023 045 011 016 042 044
014 041 047 025 018 040 065 039 024 021 036 041 013 030 022
013 093 106 017 019 027 030 042 012 021 043 013 014 020 027
019 040 029 023 016 024 015 026 011 010 026 017 012 013 014
022 042 030 040 019 015 016 011 012 009 008 012 009 017 019
022 026 018 030 020 012 017 010 008 011 007 015 008 016 034
019 018 048 029 012 054 012 008 008 009 008 012 007 016 005
022 015 057 043 126 135 122 006 005 008 007 019 010 011 008
018 008 009 019 023 093 109 128 063 052 031 010 012 009 006
017 010 010 007 067 054 106 116 067 056 011 028 005 009 006
015 010 012 014 062 076 057 055 019 024 020 006 005 013 004
016 010 008 011 039 025 020 016 011 007 008 007 006 010 003
015 009 010 010 012 011 014 009 008 007 007 005 005 008 002
014 007 008 011 007 012 010 009 007 008 007 005 005 007 003
020 011 015 019 013 017 017 013 019 013 012 013 011 009 005
020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173



2d array of numbers

Input → ? Algorithm → Output

020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
047 109 107 118 107 115 110 120 120 124 120 128 124 132 131
047 125 130 130 122 121 117 142 131 133 134 141 149 144 135
051 139 143 139 147 134 149 069 127 144 139 144 150 161 149
054 136 161 148 147 158 055 052 034 030 158 156 165 163 156
043 144 165 159 154 171 224 191 047 030 171 165 175 164 163
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173
011 091 101 105 177 039 078 060 041 026 073 102 167 208 121
011 091 094 066 094 033 199 184 139 024 060 094 125 152 134
009 068 072 072 065 031 151 171 075 028 035 072 083 109 063
013 068 074 059 057 037 161 129 062 028 035 071 072 078 056
012 042 063 055 072 033 020 067 031 022 027 082 070 073 060
011 037 064 094 091 026 025 080 066 026 023 071 070 080 060
011 060 077 082 037 023 024 147 140 038 023 037 043 076 037
013 049 076 059 032 028 174 197 182 060 021 021 121 101 062
013 059 111 072 020 078 200 211 182 061 069 059 043 086 106
007 053 057 092 023 105 189 230 210 084 034 021 017 033 091
011 061 072 018 027 054 069 068 062 023 045 011 016 042 044
014 041 047 025 018 040 065 039 024 021 036 041 013 030 022
013 093 106 017 019 027 030 042 012 021 043 013 014 020 027
019 040 029 023 016 024 015 026 011 010 026 017 012 013 014
022 042 030 040 019 015 016 011 012 009 008 012 009 017 019
022 026 018 030 020 012 017 010 008 011 007 015 008 016 034
019 018 048 029 012 054 012 008 008 009 008 012 007 016 005
022 015 057 043 126 135 122 006 005 008 007 019 010 011 008
018 008 009 019 023 093 109 128 063 052 031 010 012 009 006
017 010 010 007 067 054 106 116 067 056 011 028 005 009 006
015 010 012 014 062 076 057 055 019 024 020 006 005 013 004
016 010 008 011 039 025 020 016 011 007 008 007 006 010 003
015 009 010 010 012 011 014 009 008 007 007 005 005 008 002
014 007 008 011 007 012 010 009 007 008 007 005 005 007 003
020 011 015 019 013 017 017 013 019 013 012 013 011 009 005
020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173

2d array of numbers

an image?

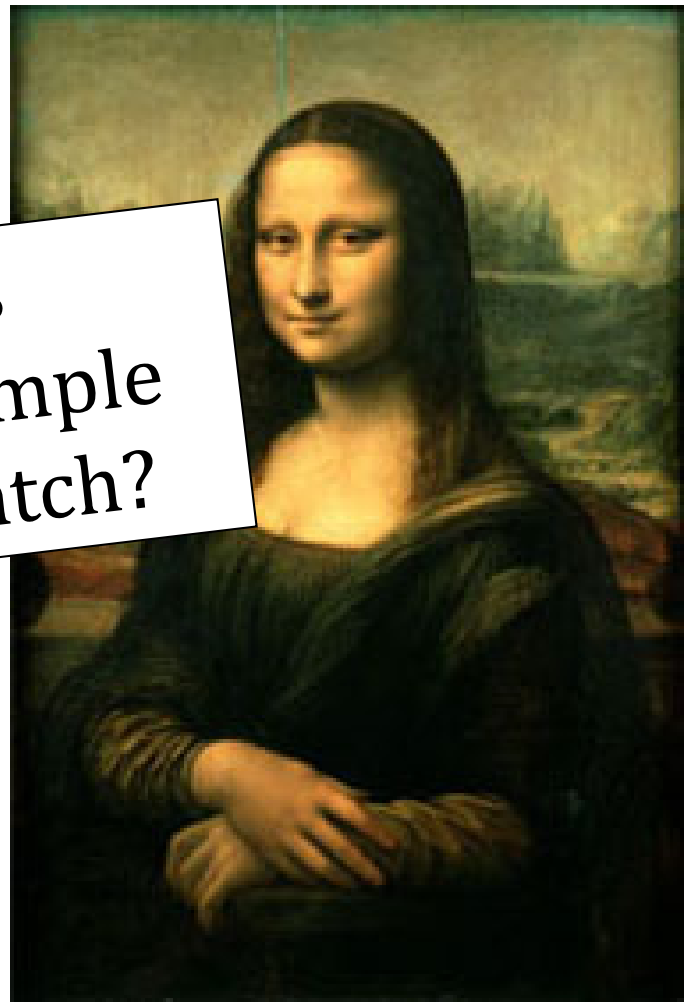
Input → ? Algorithm → Output

020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
047 109 107 118 107 115 110 120 120 124 120 128 124 132 131
047 125 130 130 122 121 117 142 131 133 134 141 149 144 135
051 139 143 139 147 134 149 069 127 144 139 144 150 161 149
054 136 161 148 147 158 055 052 034 030 158 156 165 163 156
043 144 165 159 154 171 224 191 047 030 171 165 175 164 163
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173
011 091 101 105 177 039 078 060 041 026 073 102 167 208 121
011 091 094 066 094 033 199 184 139 024 060 094 125 152 134
009 068 072 072 065 031 151 171 075 028 035 072
013 068 074 059 057 037 161 12
012 042 063 055 072 033 020 067
011 037 064 094 091 026 025 080
011 060 077 082 037 023 024 147
013 049 076 059 032 028 174 197
013 059 111 072 020 078 200 211
007 053 057 092 023 105 189 230
011 061 072 018 027 054 069 068
014 041 047 025 018 040 065 039
013 093 106 017 019 027 030 042
019 040 029 023 016 024 015 026
022 042 030 040 019 015 016 011
022 026 018 030 020 012 017 010 008 011 007 015 008 016 034
019 018 048 029 012 054 012 008 008 009 008 012 007 016 005
022 015 057 043 126 135 122 006 005 008 007 019 010 011 008
018 008 009 019 023 093 109 128 063 052 031 010 012 009 006
017 010 010 007 067 054 106 116 067 056 011 028 005 009 006
015 010 012 014 062 076 057 055 019 024 020 006 005 013 004
016 010 008 011 039 025 020 016 011 007 008 007 006 010 003
015 009 010 010 012 011 014 009 008 007 007 005 005 008 002
014 007 008 011 007 012 010 009 007 008 007 005 005 007 003
020 011 015 019 013 017 017 013 019 013 012 013 011 009 005
020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173

Why does this input/output example definitely ***not*** match?

an image?

2d array of numbers



Actual image 

Actual input: ***pixels***

207 200 199 194 203 126 106 091 113 153 192 196 190 186 175
242 194 213 254 255 114 082 109 097 089 208 206 202 194 185
254 159 187 245 248 128 197 191 166 102 248 251 233 223 246
255 159 191 245 249 123 113 047 124 149 255 253 239 143 250
255 184 191 249 249 126 022 040 115 039 250 255 255 177 252
253 197 190 233 251 054 143 209 196 014 254 254 253 228 251
252 228 176 218 251 070 176 219 200 024 250 255 255 254 195
248 244 163 227 252 091 071 185 079 021 252 251 255 254 138
253 254 169 241 253 141 053 180 005 019 253 253 255 253 201
253 250 170 242 253 138 199 193 216 146 255 253 253 253 233
254 228 164 209 235 144 188 144 161 192 230 250 253 255 252
252 240 140 215 245 119 098 086 139 127 252 234 253 253 251
253 246 169 207 235 120 157 140 188 108 246 249 235 225 255
253 230 167 207 217 113 106 069 168 098 244 246 239 207 254
253 219 170 225 253 122 144 169 124 109 243 235 233 252 252
255 221 179 239 227 123 123 069 125 122 240 249 243 232 253
221 217 180 213 243 126 143 081 181 133 252 249 244 221 210
236 216 178 208 230 225 160 118 187 198 249 249 230 220 221
229 224 183 217 216 165 209 133 193 209 222 238 236 220 214
230 223 185 240 225 212 027 187 202 239 233 234 237 225 232
221 215 189 207 220 221 003 063 209 006 204 203 232 229 216
223 217 198 162 194 215 192 106 077 255 213 219 194 215 219
224 216 162 189 200 254 217 198 209 216 206 226 186 228 219
215 222 168 202 214 201 117 084 085 111 234 190 248 218 226
219 216 198 165 213 227 014 221 197 183 178 186 216 214 222
222 224 197 146 201 204 108 220 201 210 196 223 200 233 218
226 232 177 184 175 230 214 199 146 255 134 225 205 216 181
253 254 194 192 196 168 032 193 194 196 040 198 190 207 185
045 030 167 163 210 041 206 144 129 170 086 190 161 193 191
200 205 135 143 100 174 027 175 189 113 121 150 190 209 184
202 205 138 213 088 037 164 194 194 182 076 077 107 211 181
196 208 155 063 055 068 180 200 193 160 220 082 070 210 181
161 132 058 146 048 076 172 165 218 189 186 063 004 187 185
157 080 191 119 044 025 089 115 063 192 223 146 116 186 187

020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
047 109 107 118 107 115 110 120 120 124 120 128 124 132 131
047 125 130 130 122 121 117 142 131 133 134 141 149 144 135
051 139 143 139 147 134 149 069 127 144 139 144 150 161 149
054 136 161 148 147 158 055 052 034 030 158 156 165 163 156
043 144 165 159 154 171 224 191 047 030 171 165 175 164 163
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173
011 091 101 105 177 039 078 060 041 026 073 102 167 208 121
011 091 094 066 094 033 199 184 139 024 060 094 125 152 134
009 068 072 072 065 031 151 171 075 028 035 072 083 109 063
013 068 074 059 057 037 161 129 062 028 035 071 072 078 056
012 042 063 055 072 033 020 067 031 022 027 082 070 073 060
011 037 064 094 091 026 025 080 066 026 023 071 070 080 060
011 060 077 082 037 023 024 147 140 038 023 037 043 076 037
013 049 076 059 032 028 174 197 182 060 021 021 121 101 062
013 059 111 072 020 078 200 211 182 061 069 059 043 086 106
007 053 057 092 023 105 189 230 210 084 034 021 017 033 091
011 061 072 018 027 054 069 068 062 023 045 011 016 042 044
014 041 047 025 018 040 065 039 024 021 036 041 013 030 022
013 093 106 017 019 027 030 042 012 021 043 013 014 020 027
019 040 029 023 016 024 015 026 011 010 026 017 012 013 014
022 042 030 040 019 015 016 011 012 009 008 012 009 017 019
022 026 018 030 020 012 017 010 008 011 007 015 008 016 034
019 018 048 029 012 054 012 008 008 009 008 012 007 016 005
022 015 057 043 126 135 122 006 005 008 007 019 010 011 008
018 008 009 019 023 093 109 128 063 052 031 010 012 009 006
017 010 010 007 067 054 106 116 067 056 011 028 005 009 006
015 010 012 014 062 076 057 055 019 024 020 006 005 013 004
016 010 008 011 039 025 020 016 011 007 008 007 006 010 003
015 009 010 010 012 011 014 009 008 007 007 005 005 008 002
014 007 008 011 007 012 010 009 007 008 007 005 005 007 003
020 011 015 019 013 017 017 013 019 013 012 013 011 009 005
020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173

207 200 191 194 203 131 100 078 093 145 192 196 190 186 175
242 194 221 254 255 126 063 061 052 060 208 206 202 194 185
254 176 211 255 250 126 116 086 070 064 253 252 233 239 253
251 180 203 255 251 123 075 030 070 072 255 253 239 156 255
255 206 194 255 246 141 019 021 035 026 250 255 255 184 255
253 219 190 239 248 119 061 095 065 011 254 254 253 231 255
252 247 181 218 251 142 066 142 080 026 250 255 255 255 204
248 251 167 227 252 146 047 073 047 021 252 251 255 255 145
253 255 170 255 253 147 040 094 000 020 253 253 255 255 203
253 250 170 255 253 144 092 077 106 057 255 253 253 255 238
254 228 173 214 235 145 083 073 058 089 230 250 253 252 253
252 245 140 215 245 125 038 027 044 042 252 234 253 253 252
255 251 169 214 235 110 075 065 060 052 246 249 235 225 255
255 235 167 216 217 105 048 038 046 046 244 246 239 207 254
254 219 170 247 253 104 121 057 042 047 243 235 233 252 254
255 221 179 255 227 099 058 029 034 052 240 249 243 232 253
221 217 180 213 243 081 052 029 055 023 250 249 244 221 210
236 216 178 208 230 149 043 037 056 075 241 249 230 220 221
229 224 183 213 121 036 076 034 069 073 116 248 236 220 214
230 223 185 178 098 070 010 070 097 098 108 101 236 223 232
221 215 190 084 122 070 011 018 075 013 076 079 146 227 216
223 217 120 033 059 071 042 044 013 155 060 070 076 213 219
224 216 030 082 075 139 076 083 099 114 076 133 057 228 219
215 222 033 092 067 063 031 023 070 025 125 055 136 224 226
219 216 076 044 069 119 011 102 078 055 043 054 111 226 222
222 224 063 046 065 080 101 090 054 104 109 123 052 216 218
226 232 086 080 042 123 083 054 031 148 045 072 067 061 164
253 255 072 058 079 056 006 046 056 061 026 085 082 048 127
046 033 080 045 105 027 071 067 054 036 035 059 060 048 120
174 173 047 042 019 055 011 044 095 033 031 052 054 061 118
173 175 039 127 038 024 053 054 050 051 036 038 022 063 126
170 160 058 020 027 027 045 074 057 035 124 036 016 072 124
107 054 027 044 020 035 062 039 115 048 051 046 005 051 117
098 006 045 033 030 027 036 054 025 072 112 044 029 060 115

Actual output: ***contents***

it's a woman, smiling?

Actual image 

Actual input: ***pixels***

207 200 199 194 203 126 106 091 113 153 192 196 190 186 175
242 194 213 254 255 114 082 109 097 089 208 206 202 194 185
254 159 187 245 248 128 197 191 166 102 248 251 233 223 246
255 159 191 245 249 123 113 047 124 149 255 253 239 143 250
255 184 191 249 249 126 022 040 115 039 250 255 255 177 252
253 197 190 233 251 054 143 209 196 014 254 254 253 228 251
252 228 176 218 251 070 176 219 200 024 250 255 255 254 195
248 244 163 227 252 091 071 185 079 021 252 251 255 254 138
253 254 169 241 253 141 053 180 005 019 253 253 255 253 201
253 250 170 242 253 138 199 193 216 146 255 253 253 253 233
254 228 164 209 235 144 188 144 161 192 230 250 253 255 252
252 240 140 215 245 119 098 086 139 127 252 234 253 253 251
253 246 169 207 235 120 157 140 188 108
253 230 167 207 217 113
253 219
255
221
236
229
230
221 215 189 207 220 221 003 063 209 006 204 203 232
223 217 198 162 194 215 192 106 077
224 216 162 189 200 254 217
215 222 168 202 214 201 117
219 216 198 165 213 227 014
222 224 197 146 201 204 108
226 232 177 184 175 230 214
253 254 194 192 196 168 032
045 030 167 163 210 041 206 144 129 170 086 190 161 193 191
200 205 135 143 100 174 027 175 189 113 121 150 190 209 184
202 205 138 213 088 037 164 194 194 182 076 077 107 211 181
196 208 155 063 055 068 180 200 193 160 220 082 070 210 181
161 132 058 146 048 076 172 165 218 189 186 063 004 187 185
157 080 191 119 044 025 089 115 063 192 223 146 116 186 187

020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
047 109 107 118 107 115 110 120 120 124 120 128 124 132 131
047 125 130 130 122 121 117 142 131 133 134 141 149 144 135
051 139 143 139 147 134 149 069 127 144 139 144 150 161 149
054 136 161 148 147 158 055 052 034 030 158 156 165 163 156
043 144 165 159 154 171 224 191 047 030 171 165 175 164 163
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173
011 091 101 105 177 039 078 060 041 026 073 102 167
011 091 094 066 094 033 199 184 139
009 068 072 072 065
013
007 053 057 092 023 105 189 230 210 084 034 021
011 061 072 018 027 054 069 068
014 041 047 025
018 008 009 019 023 093 109 128 063 052 031 010 012 009 006
017 010 010 007 067 054 106 116 067 056 011 028 005 009 006
015 010 012 014 062 076 057 055 019 024 020 006 005 013 004
016 010 008 011 039 025 020 016 011 007 008 007 006 010 003
015 009 010 010 012 011 014 009 008 007 007 005 005 008 002
014 007 008 011 007 012 010 009 007 008 007 005 005 007 003
020 011 015 019 013 017 017 013 019 013 012 013 011 009 005
020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
025 161 174 172 167 049 200 193 112 028 120 169 173 177 173

207 200 191 194 203 131 100 078 093 145 192 196 190 186 175
242 194 221 254 255 126 063 061 052 060 208 206 202 194 185
254 176 211 255 250 126 116 086 070 064 253 252 233 239 253
251 180 203 255 251 123 075 030 070 072 255
255 206 194 255 246 141 019
253 219 190
106 057 255 253 253 255 238
214 235 145 083 073 058 089 230 250 253 252 253
252 245 140 215 245 125 038 027 044 042 252 234 253 253 252
255 251 169 214 235 110 075 065 060 052 246 249 235 225 255
255 235 167 216 217 105 048
254 219 170
040 249 243 232 253
249 244 221 210
249 230 220 221
073 116 248 236 220 214
070 010 070 097 098 108 101 236 223 232
190 084 122 070 011 018 075 013 076 079 146 227 216
223 217 120 033 059 071 042 044 013 155 060 070 076 213 219
224 216 030 082 075 139 076 083 099 114 076 133 057 228 219
215 222 033 092 067 063 031 023 070 025 125 055 136 224 226
219 216 076 044 069 119 011 102 078 055 043 054 111 226 222
222 224 063 046 065 080 101 090 054 104 109 123 052 216 218
226 232 086 080 042 123 083 054 031 148 045 072 067 061 164
253 255 072 058 079 056 006 046 056 061 026 085 082 048 127
046 033 080 045 105 027 071 067 054 036 035 059 060 048 120
174 173 047 042 019 055 011 044 095 033 031 052 054 061 118
173 175 039 127 038 024 053 054 050 051 036 038 022 063 126
170 160 058 020 027 027 045 074 057 035 124 036 016 072 124
107 054 027 044 020 035 062 039 115 048 051 046 005 051 117
098 006 045 033 030 027 036 054 025 072 112 044 029 060 115

Wonderfully, this problem is **being** solved...

but why is it so challenging?

Actual output: ***contents***

it's a woman, smiling?

What's red?

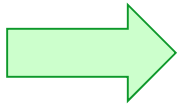
Goal: a coke-can collecting robot…

What's red?

red > 200

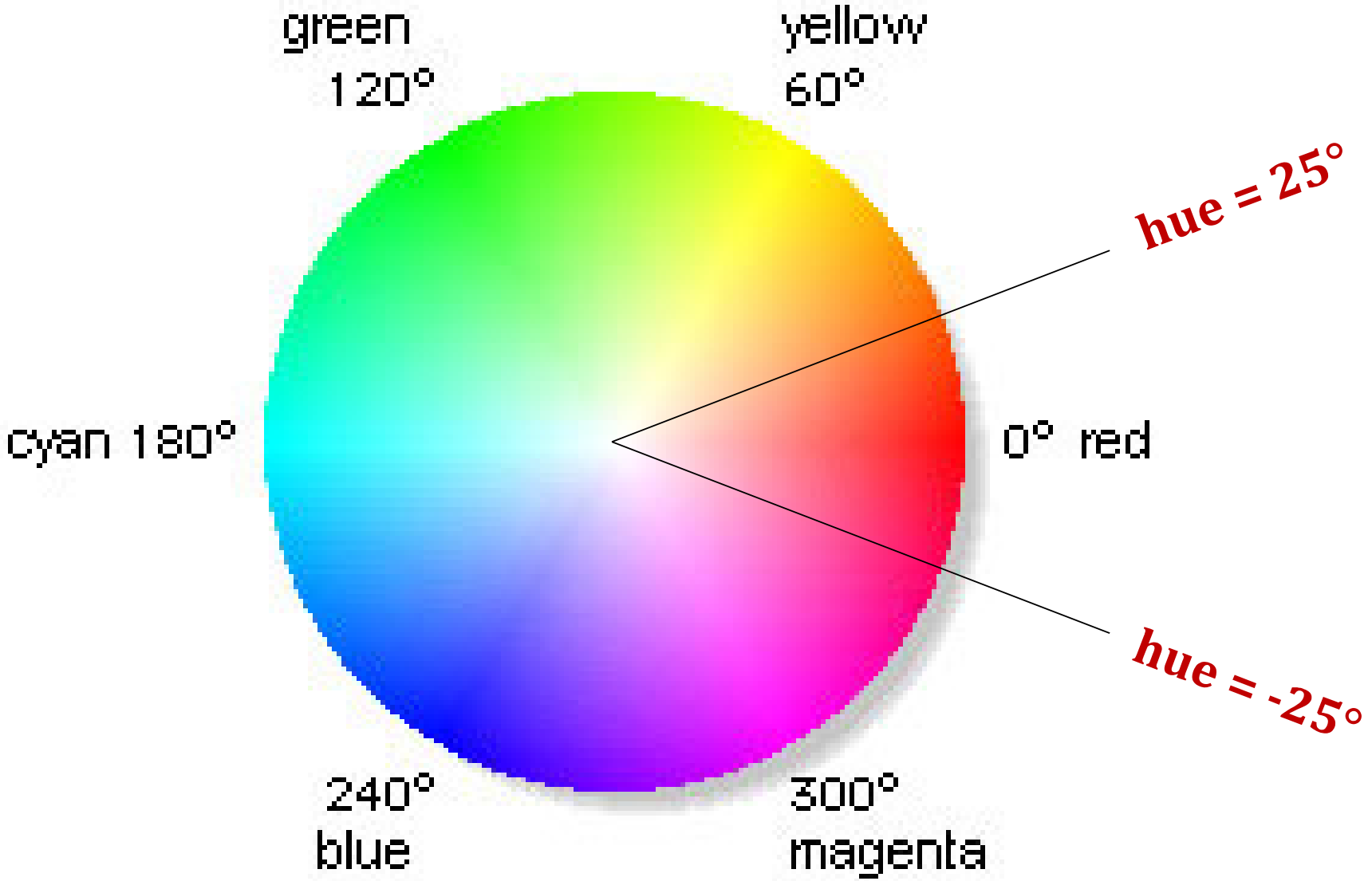
not good enough…

What's red?

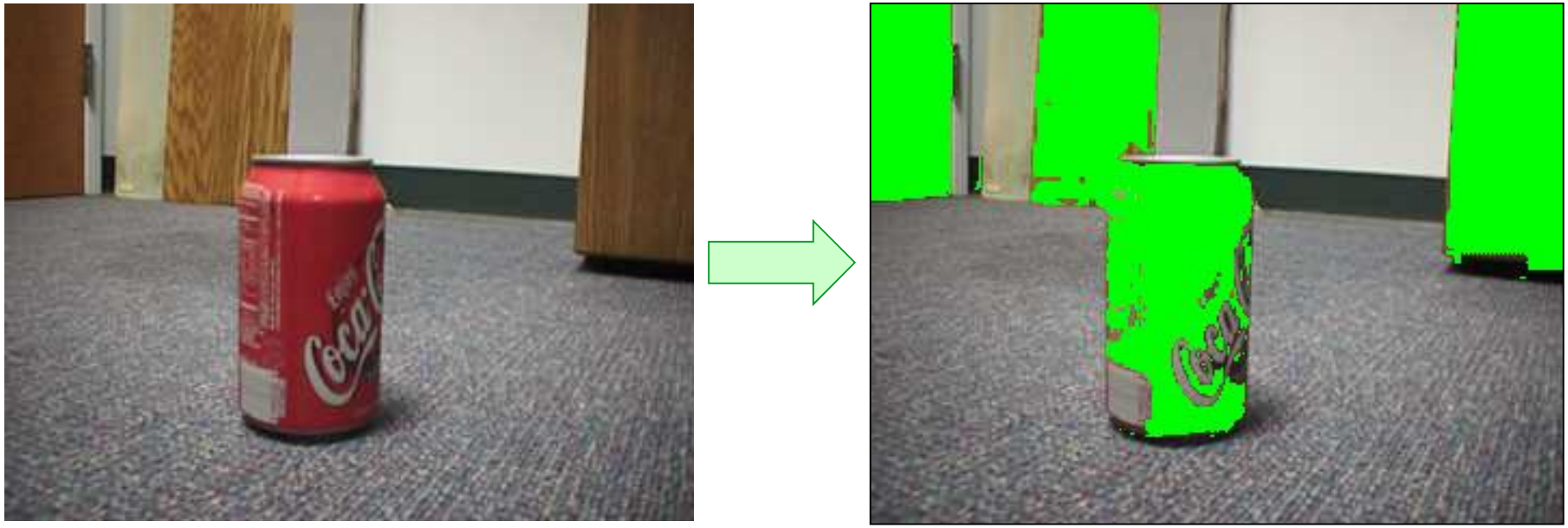


| hue | < 25

Hue and saturation



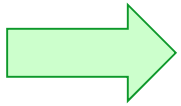
What's red?



| hue | < 25

not good enough…

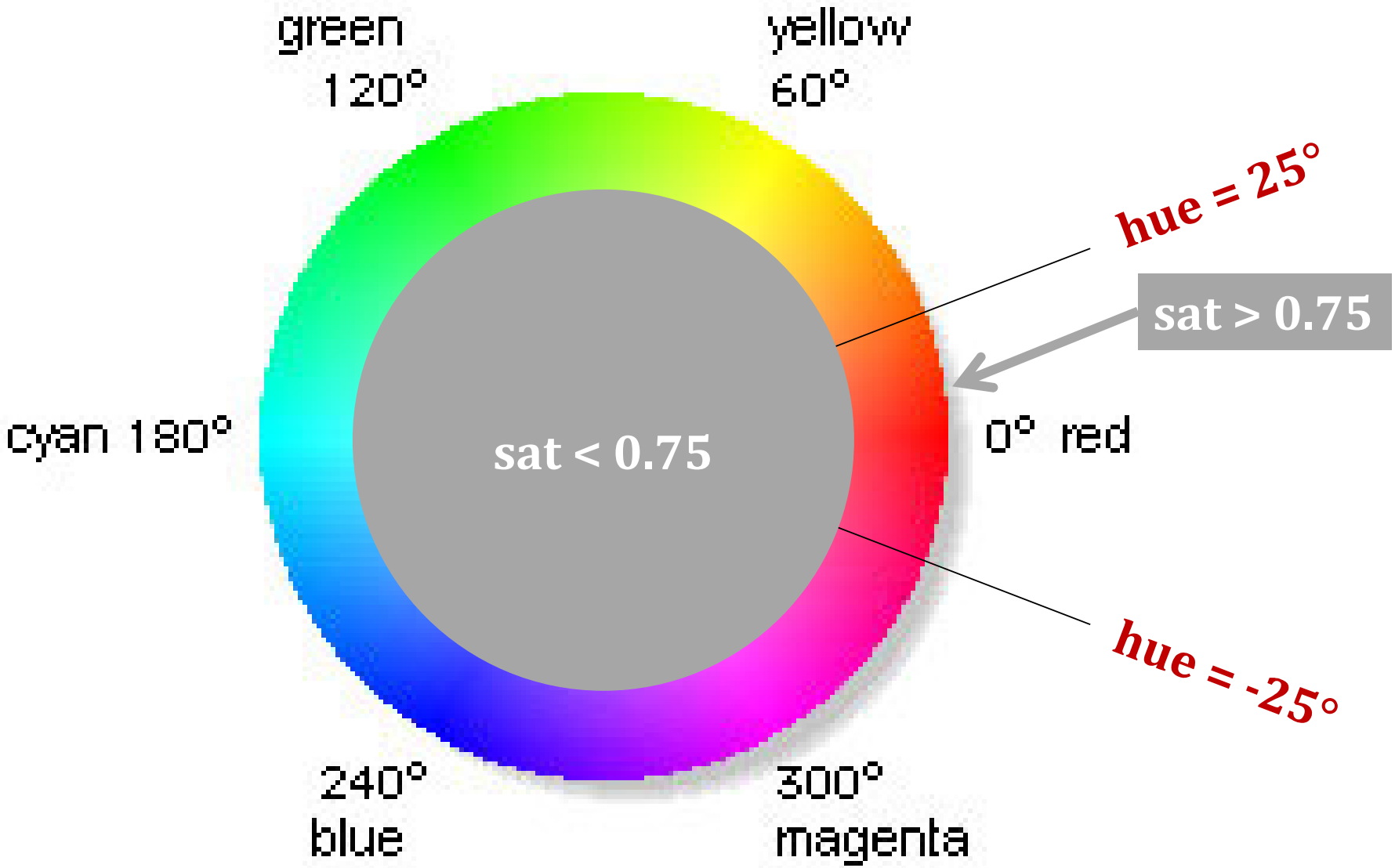
What's red?



| hue | < 25

saturation > 0.75

Hue and saturation



What's red?

Aargh!

| hue | < 25

saturation > 0.75

The door is still matched, too… ***why?***

What's red?

Aargh!

Remarkably, this problem is, in a sense, ***our own*** computers' fault...!

| hue | < 25

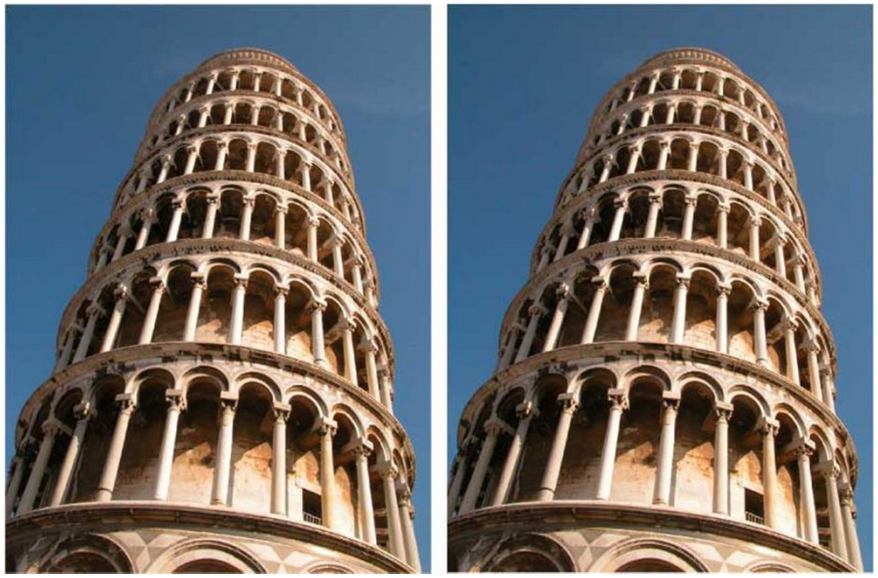
uration > 0.75

door is still matched, too... ***why?***

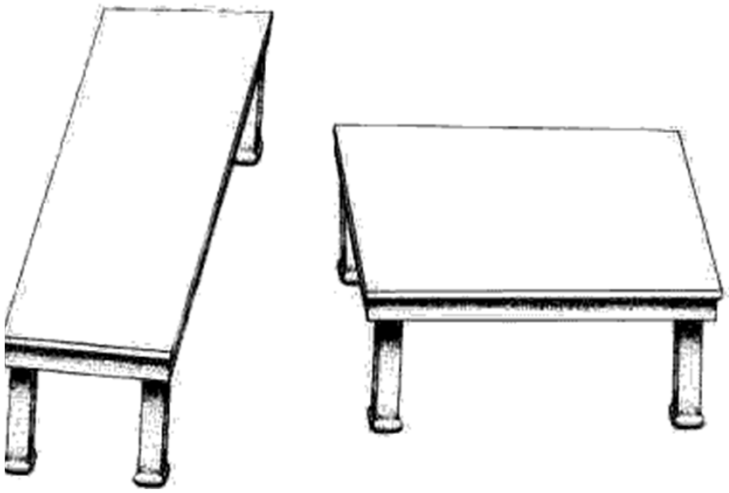
Quiz

*Illusions? What **computations** is your brain doing to cause them?* **Why?**

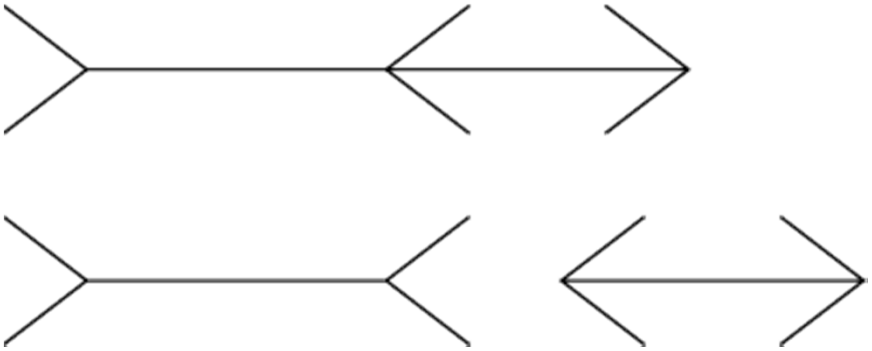
two tones



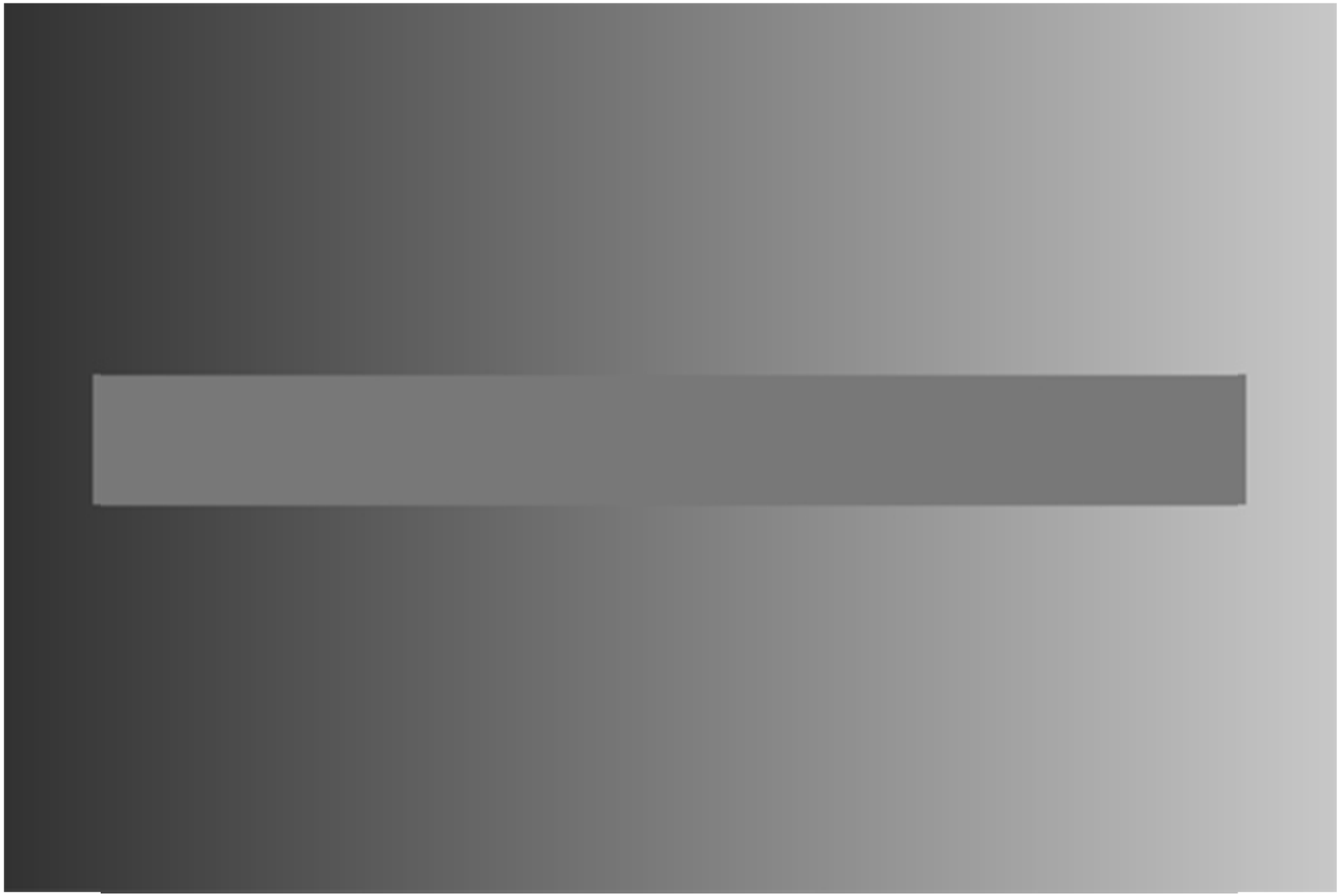
two towers



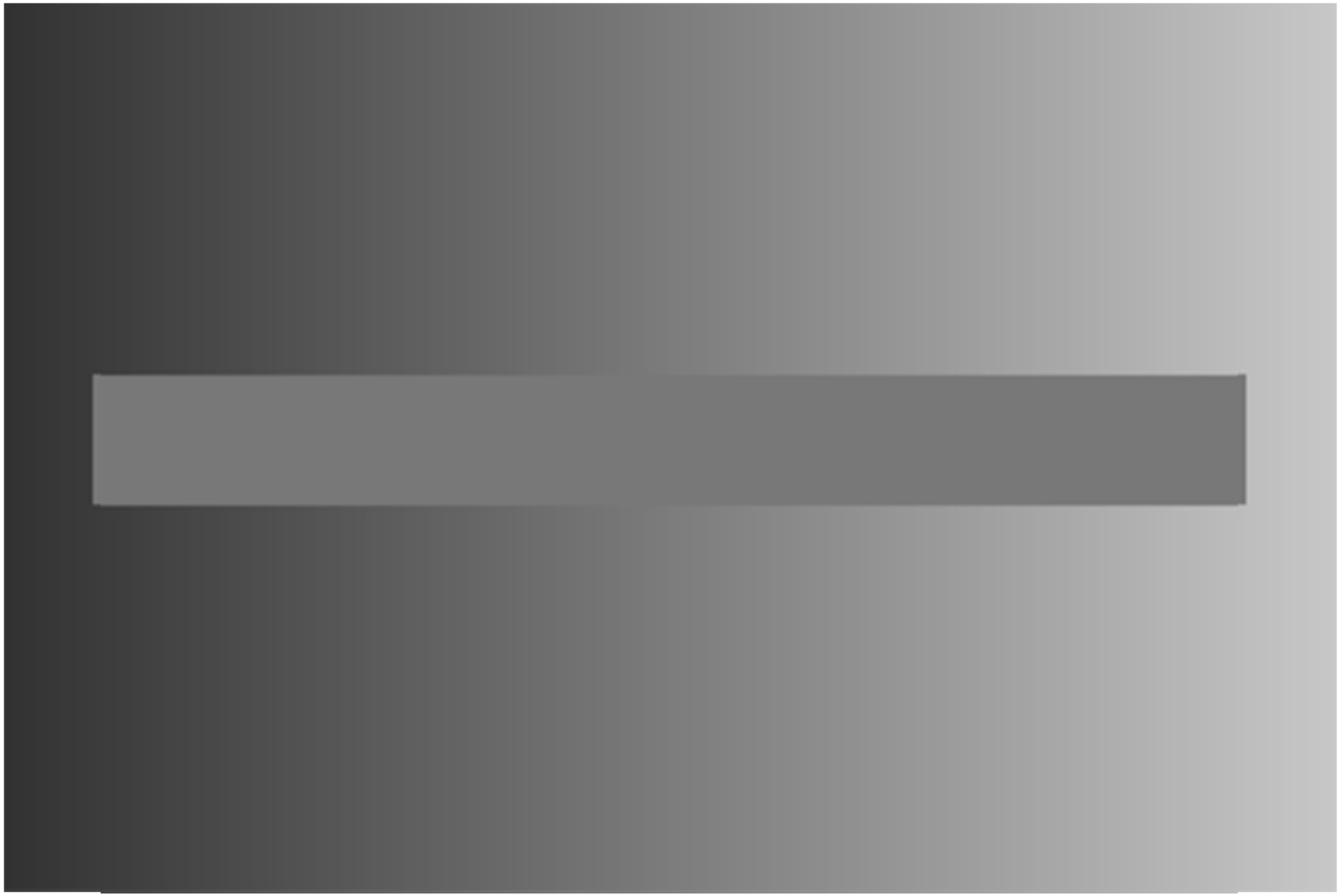
two tables



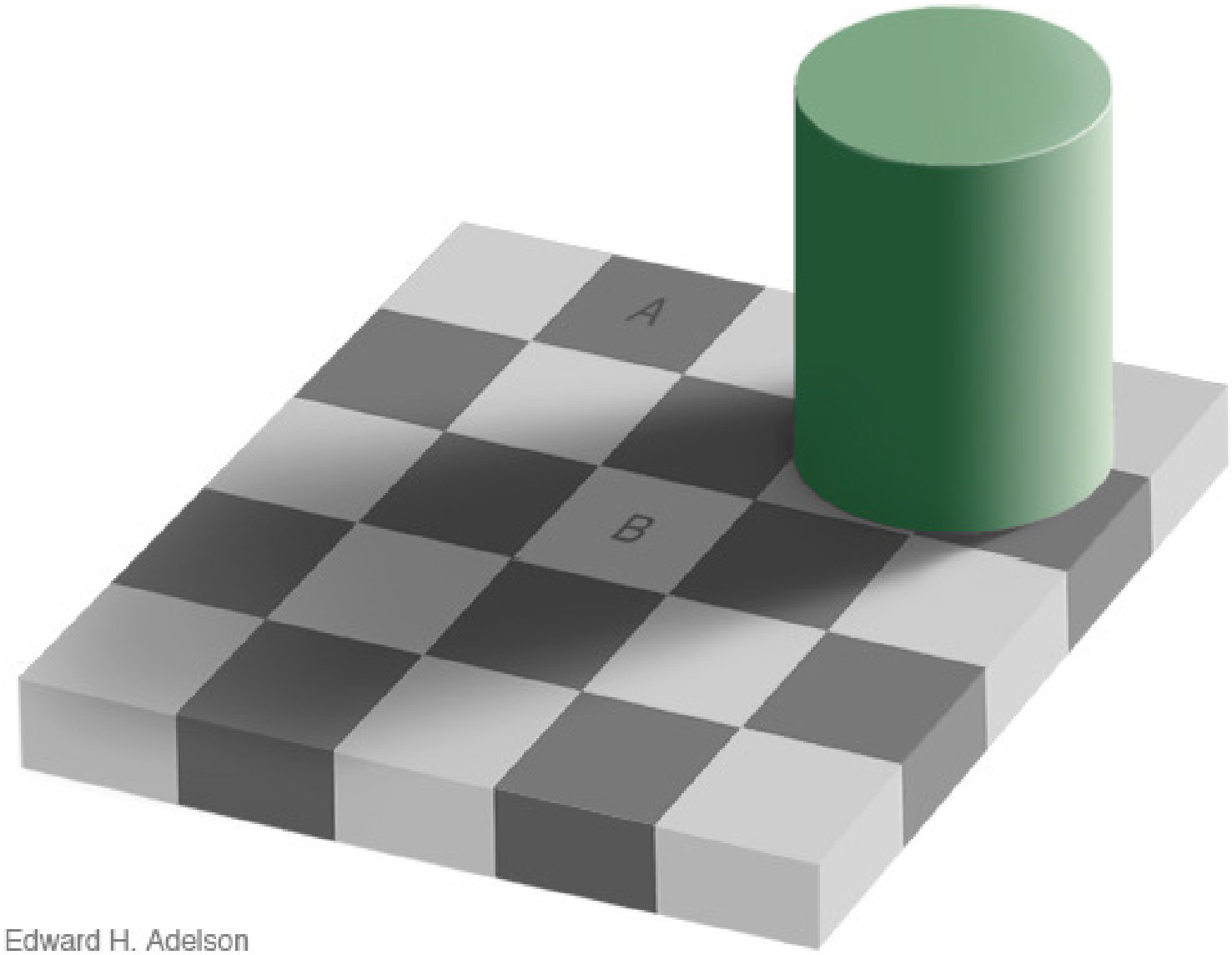
two segments



2 layers

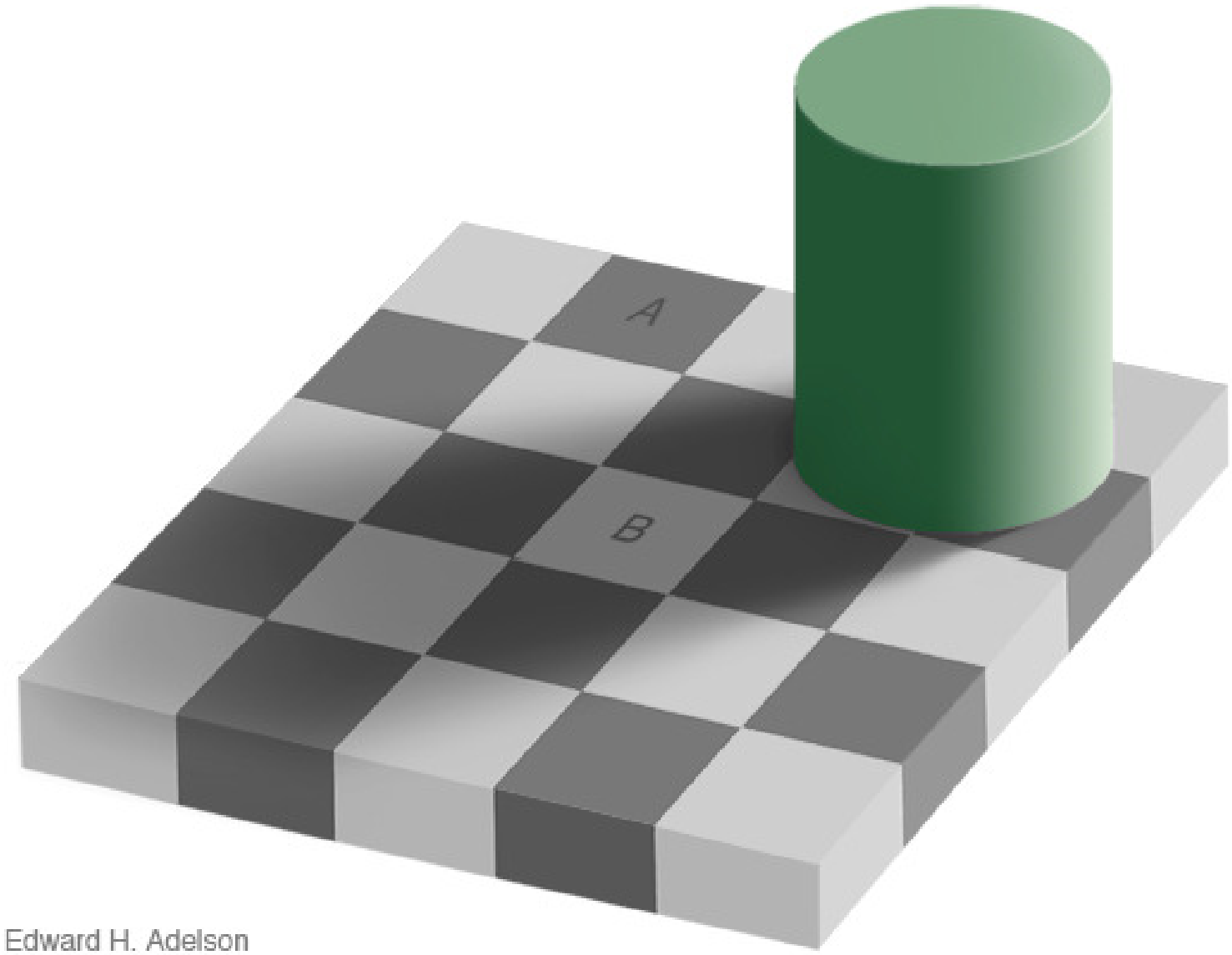


2 layers



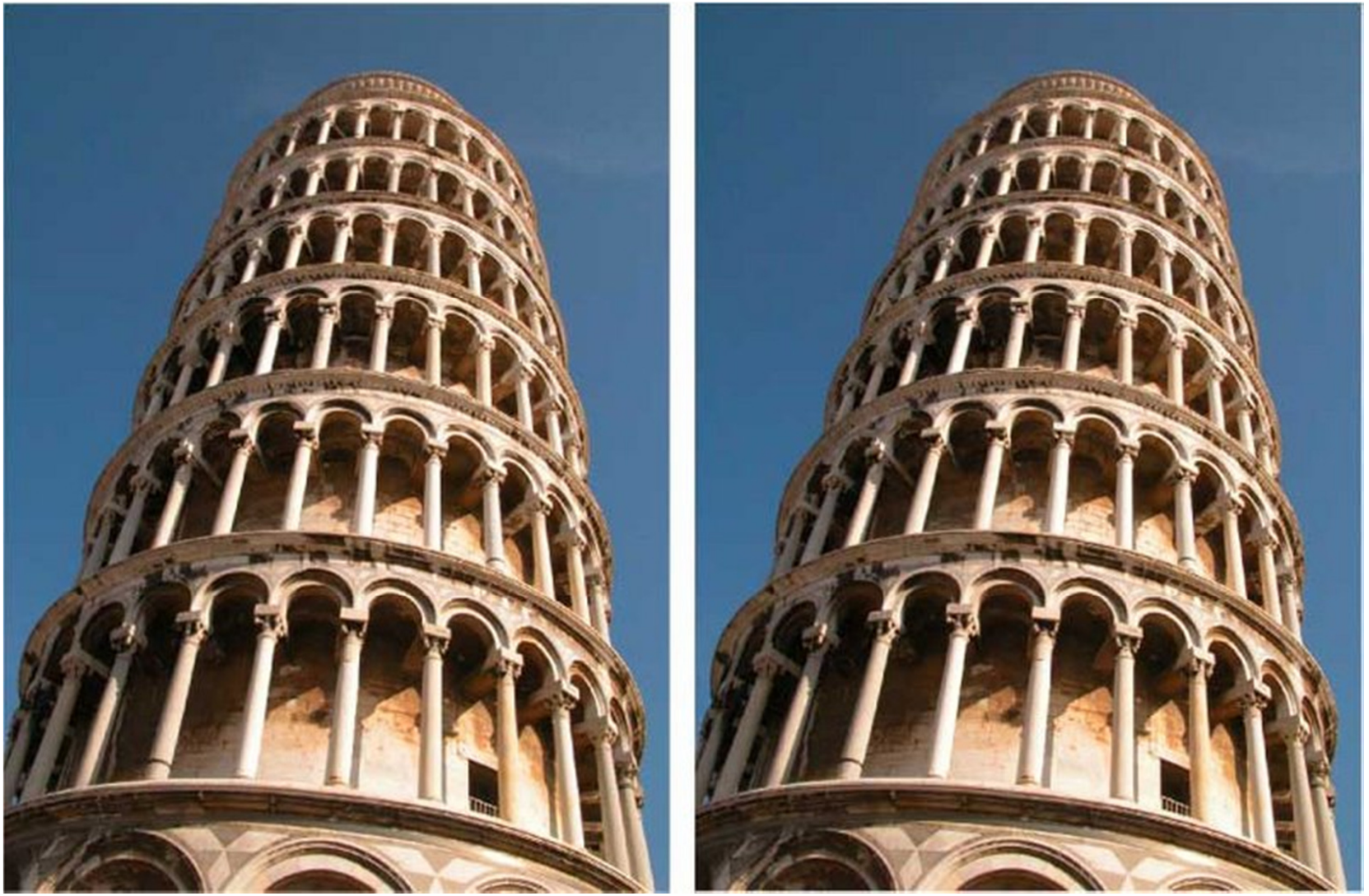
Edward H. Adelson

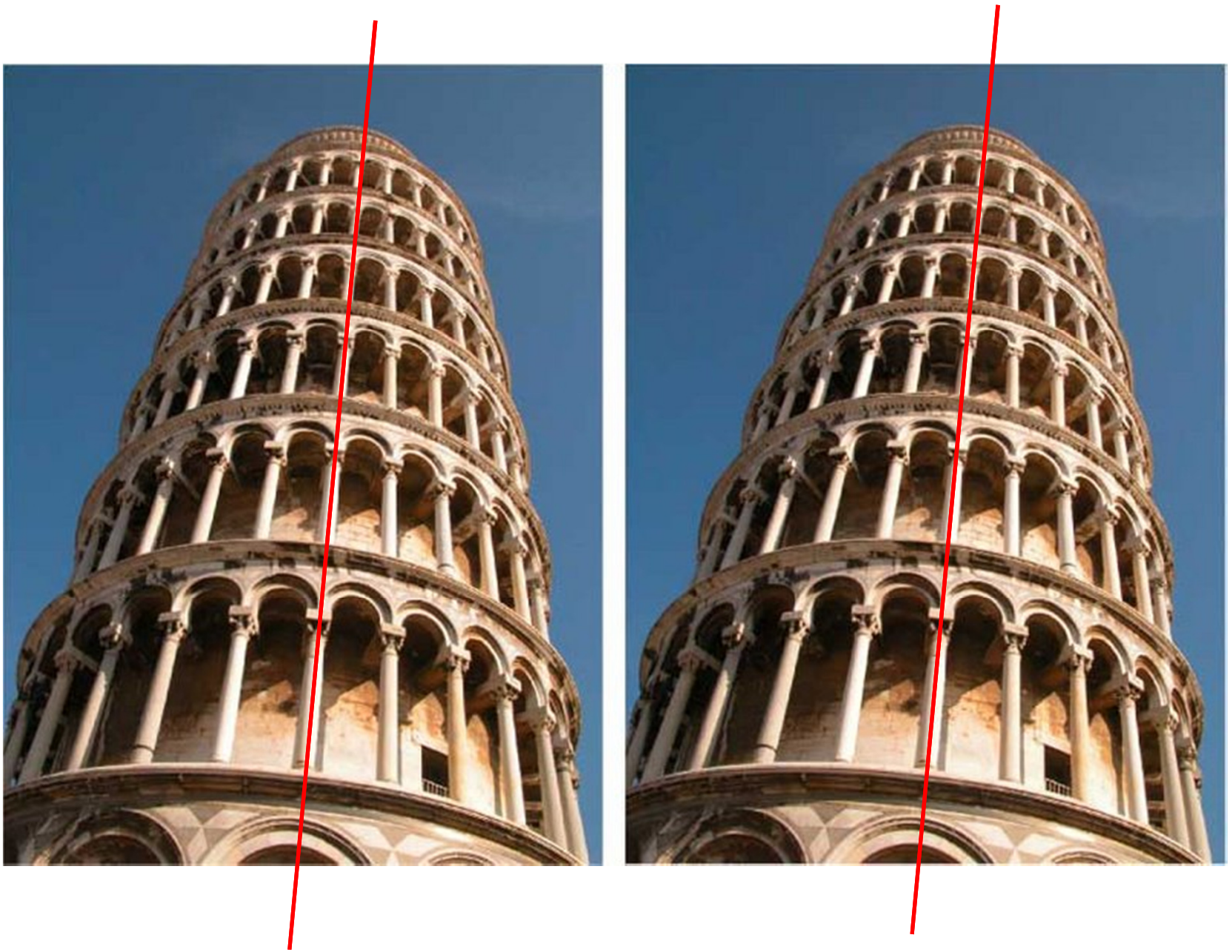
2 layers

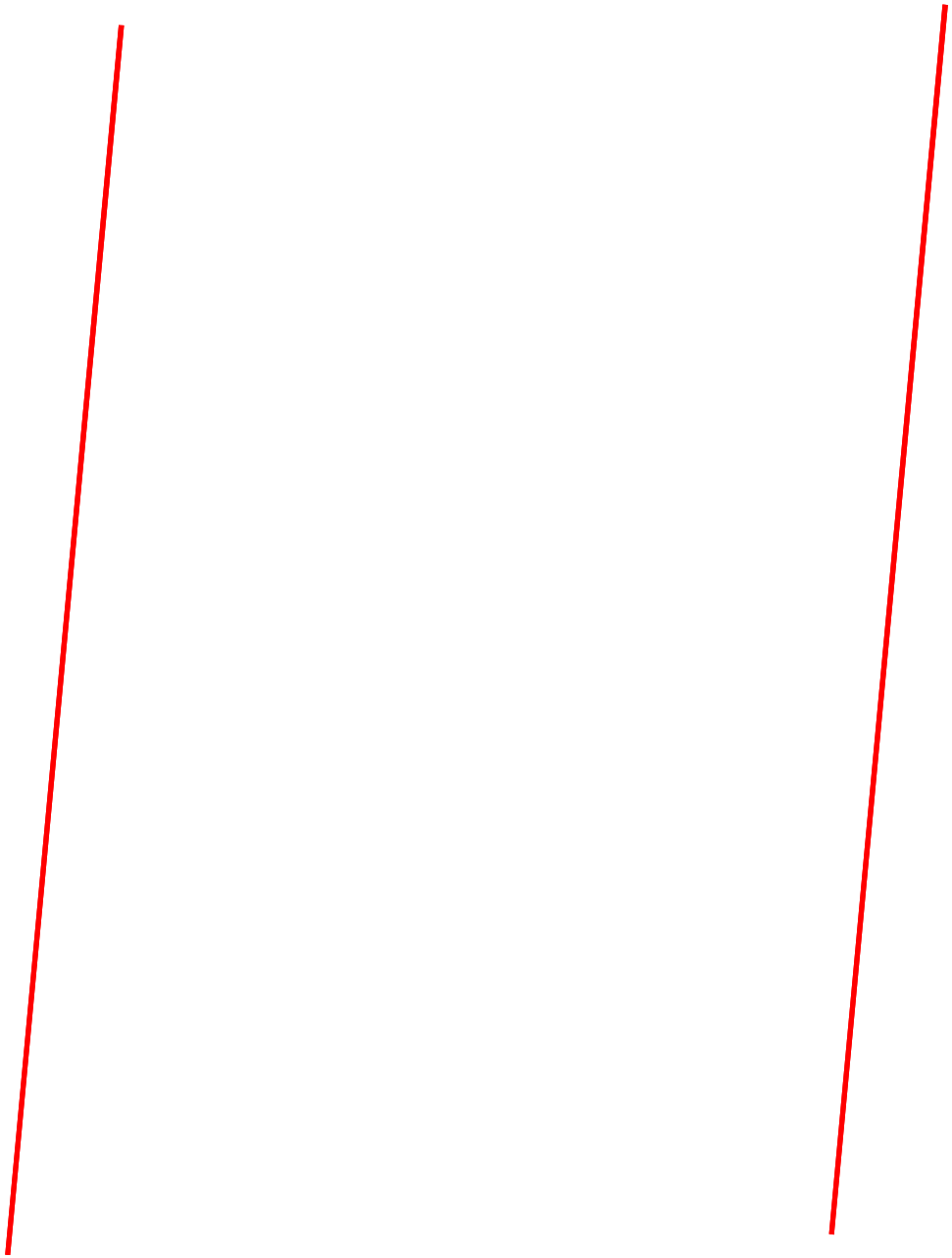


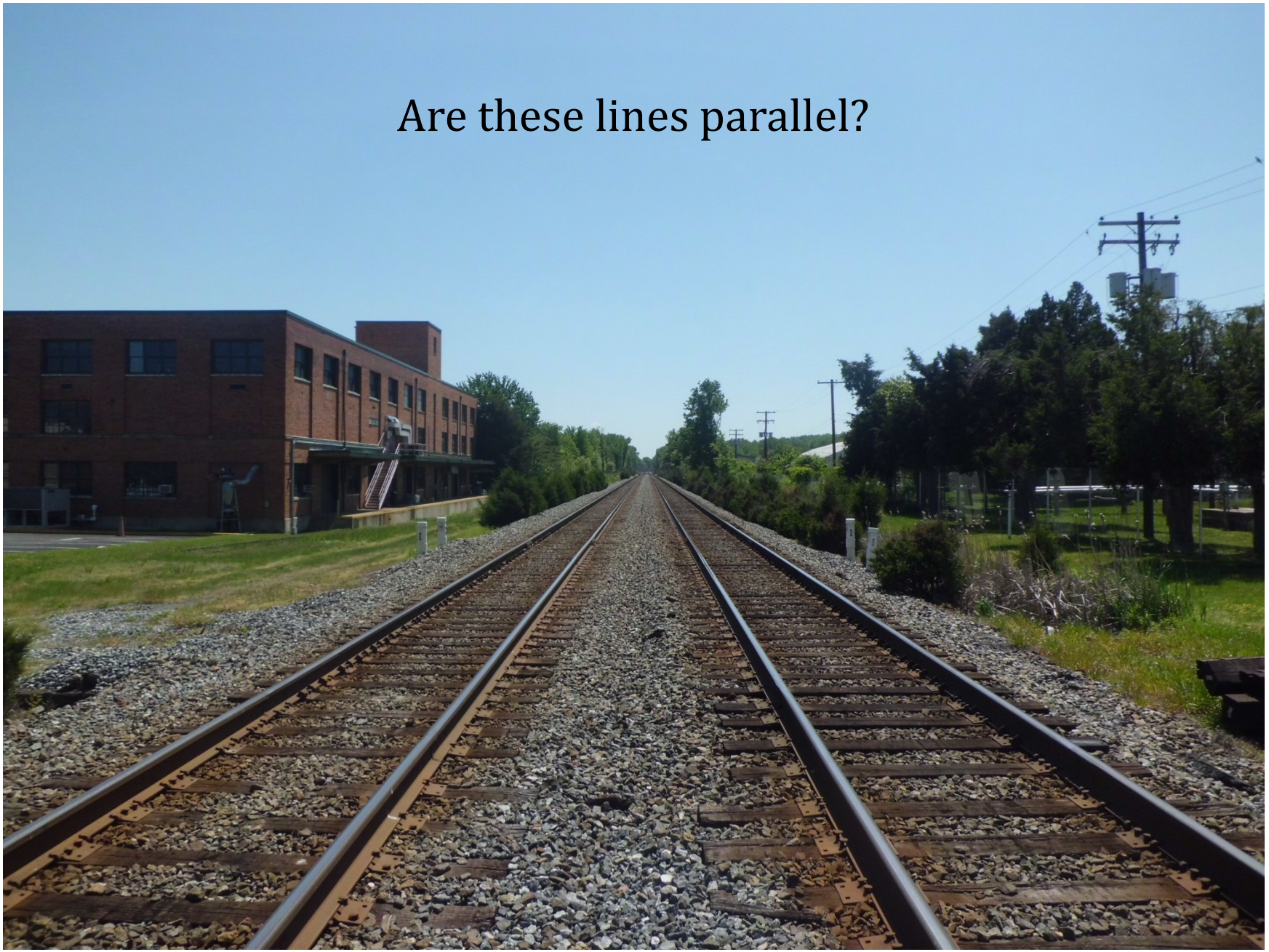
Edward H. Adelson

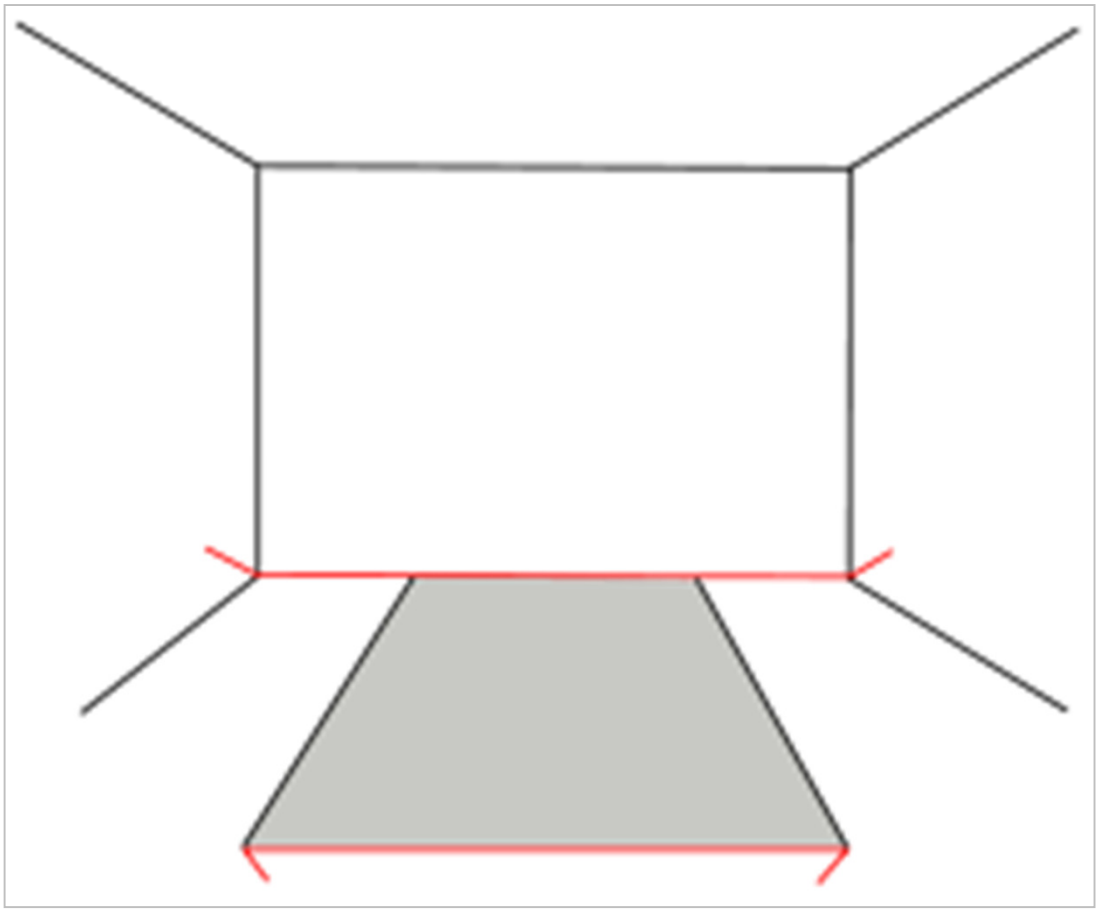
2 layers

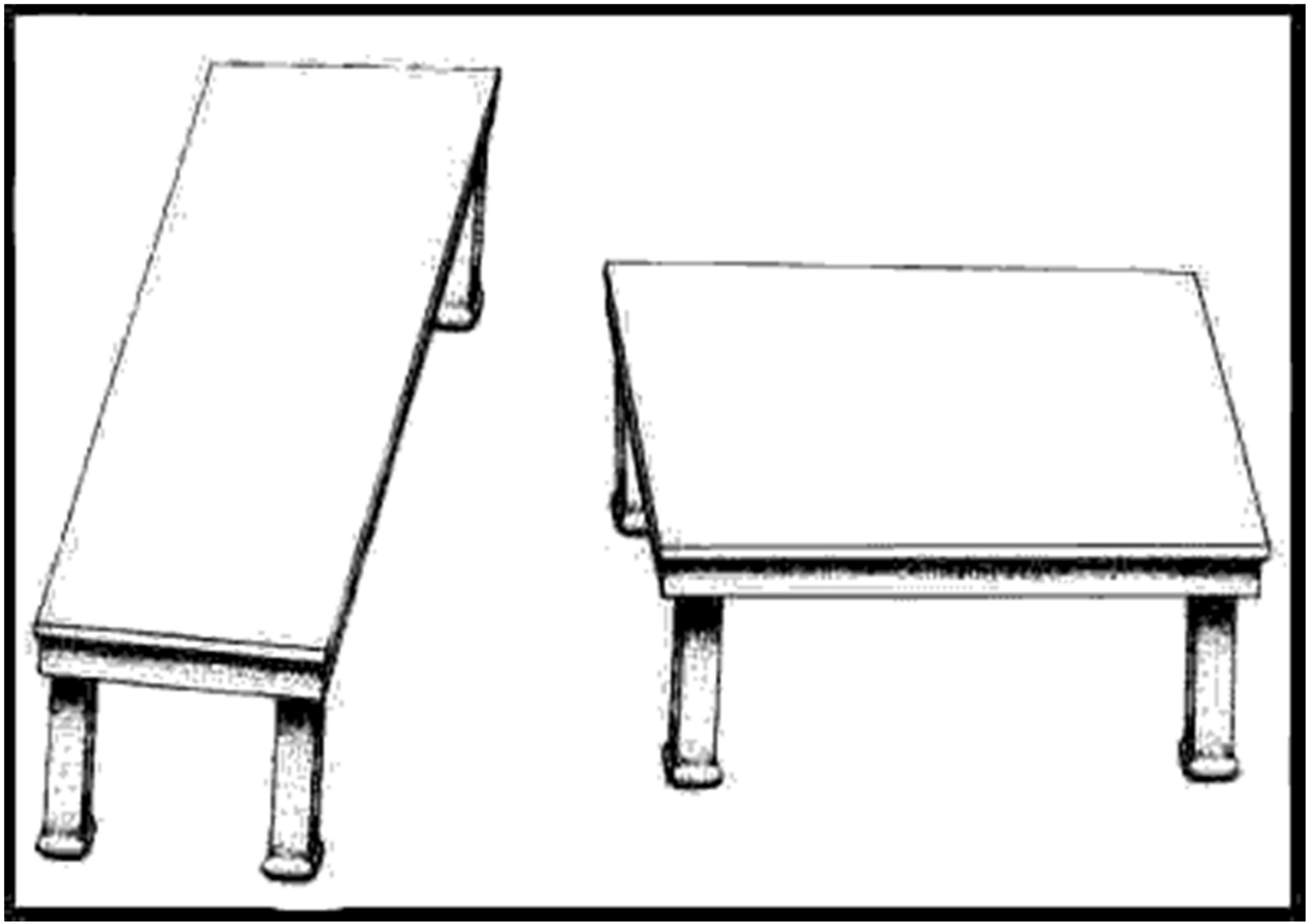


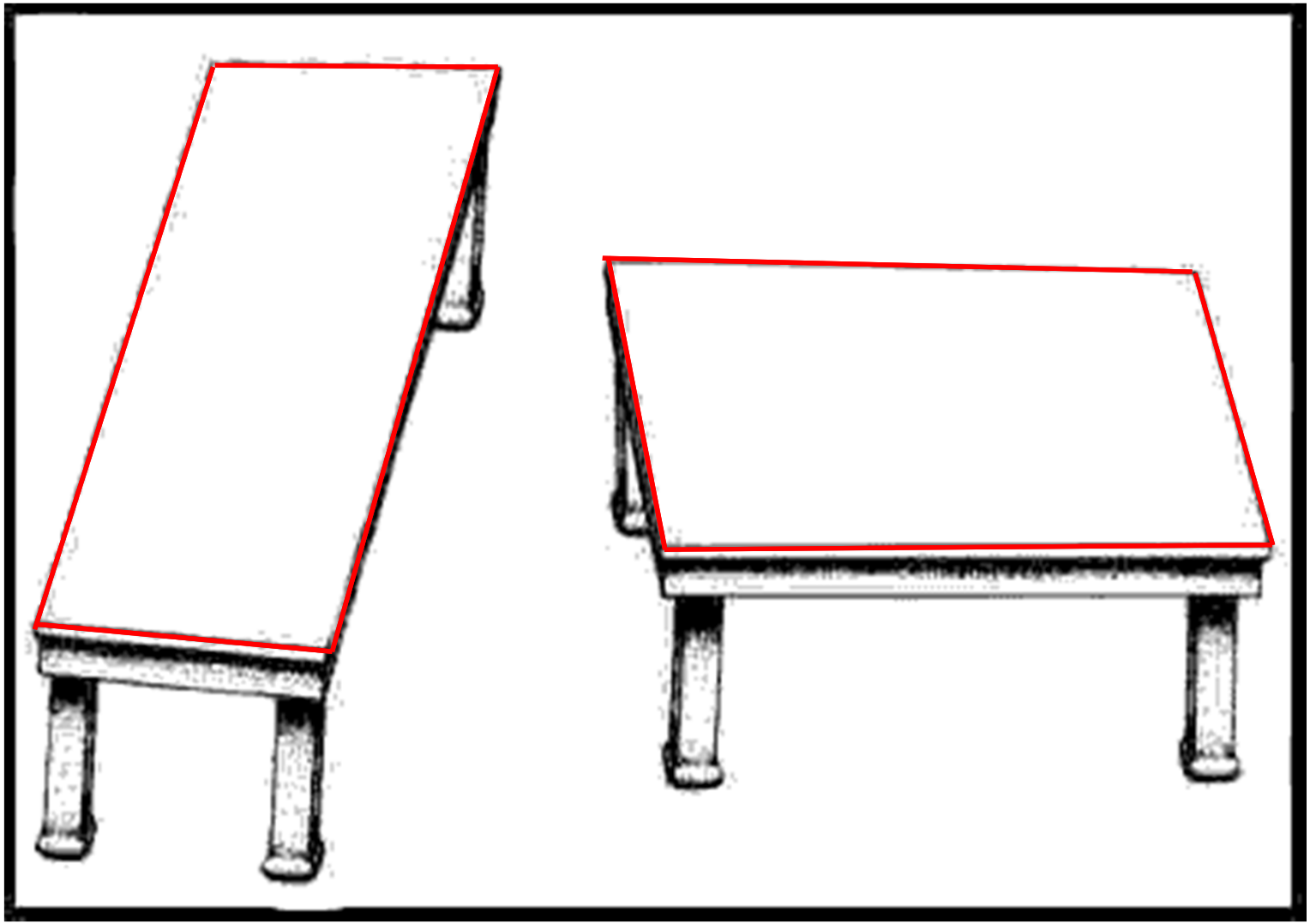


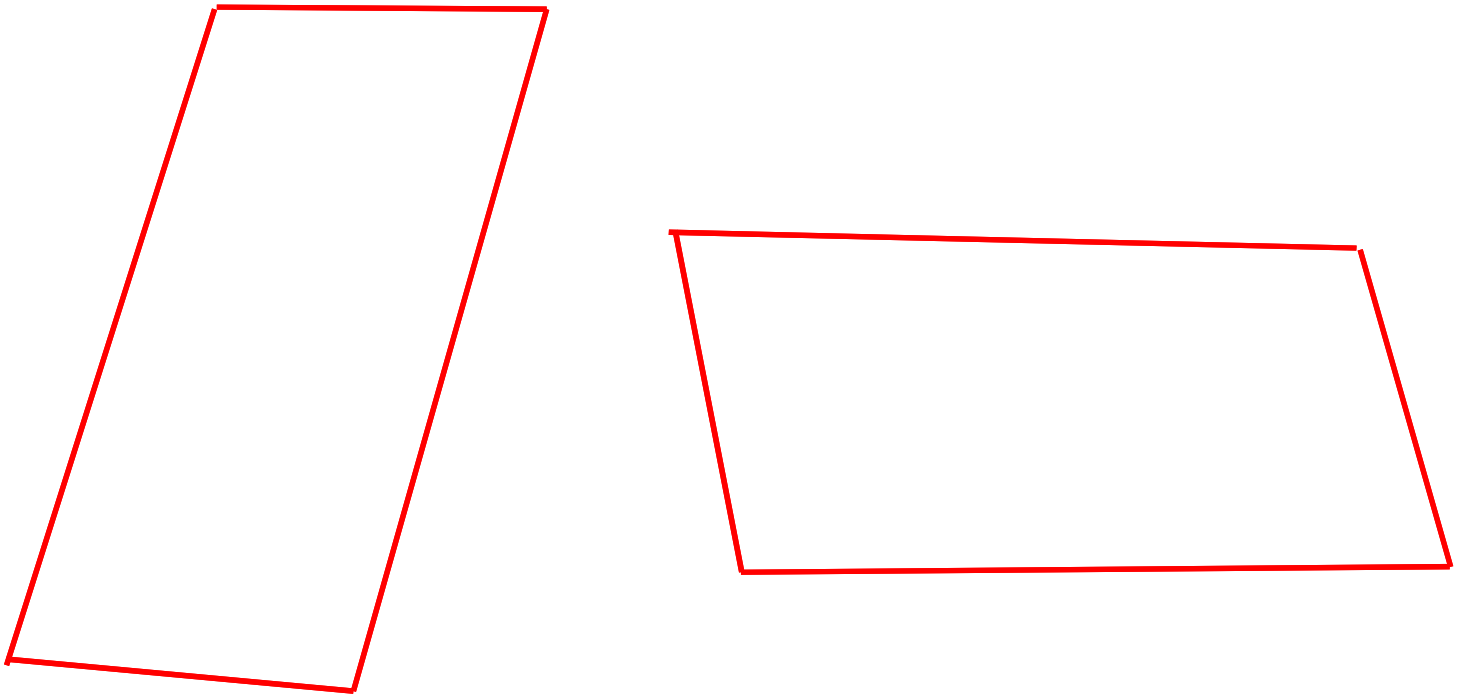



Are these lines parallel?





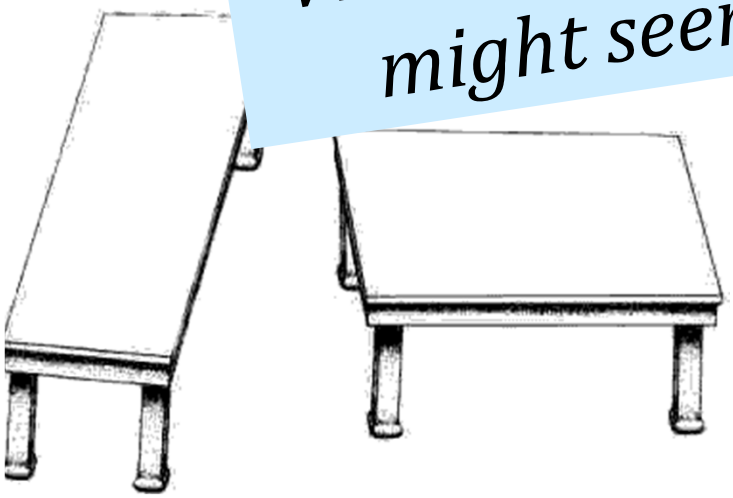


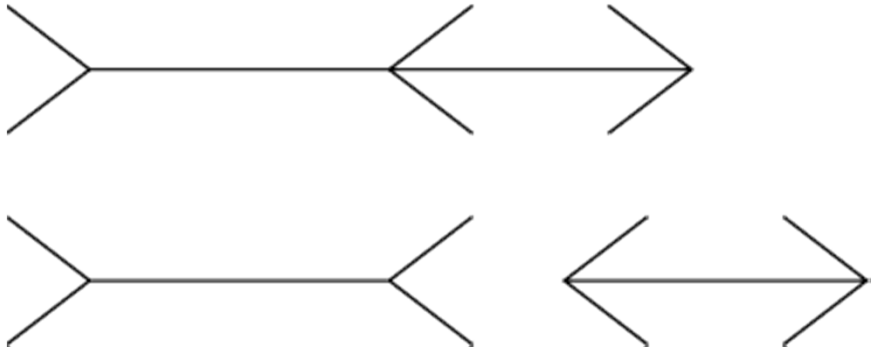


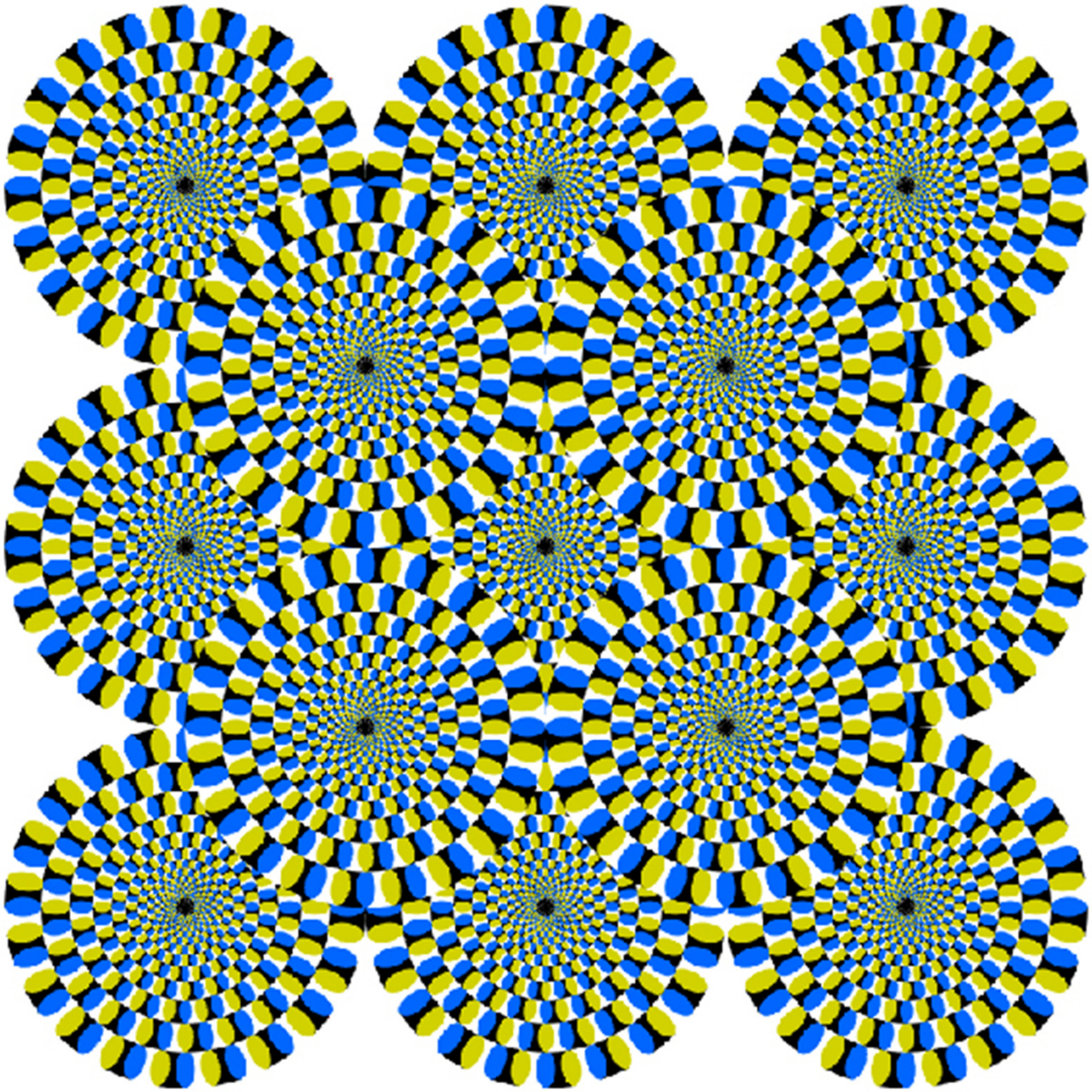
Quiz

Illusions? What computations is your brain doing to cause them?

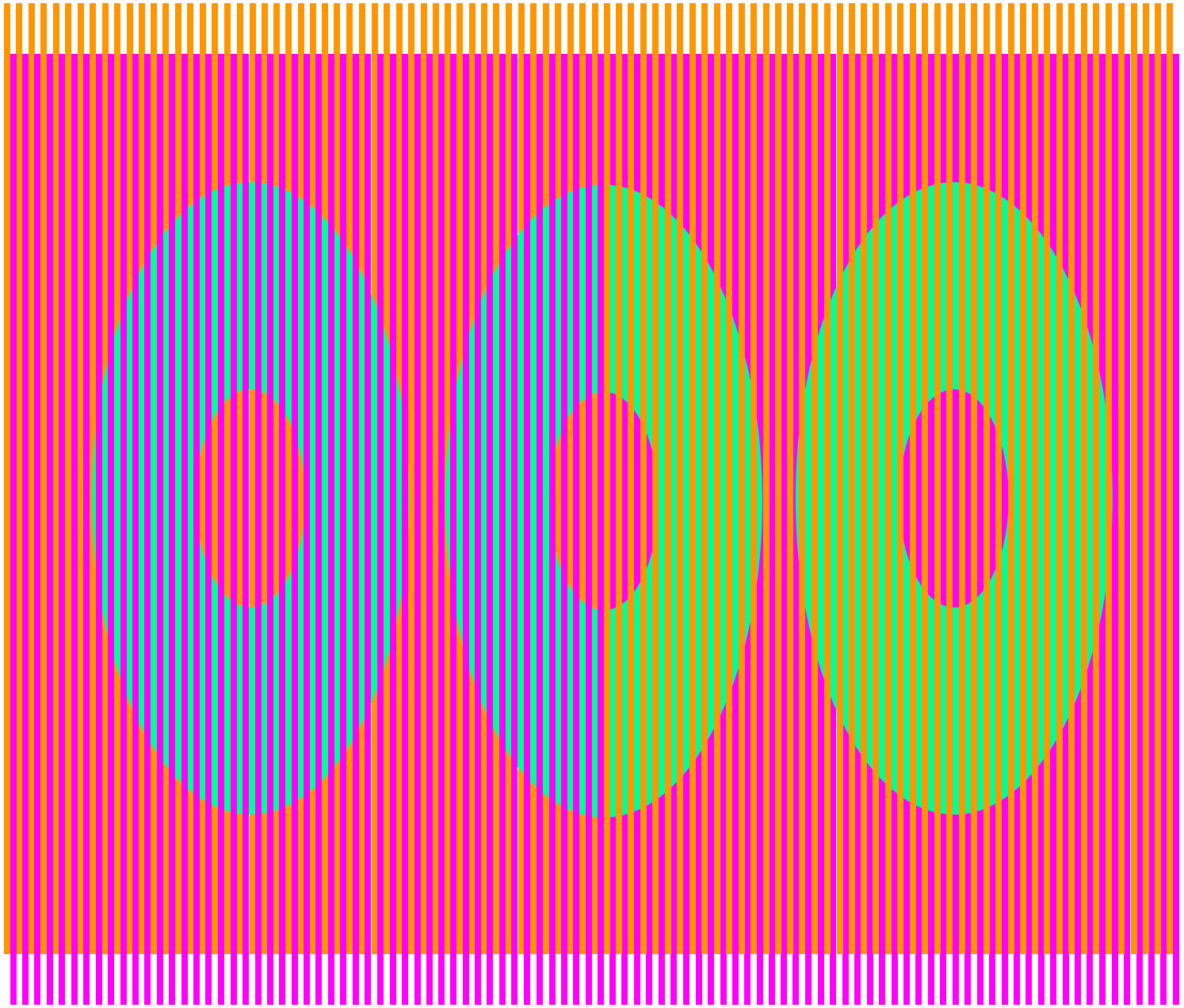
Vision is more challenging than it might seem on first "glance"!



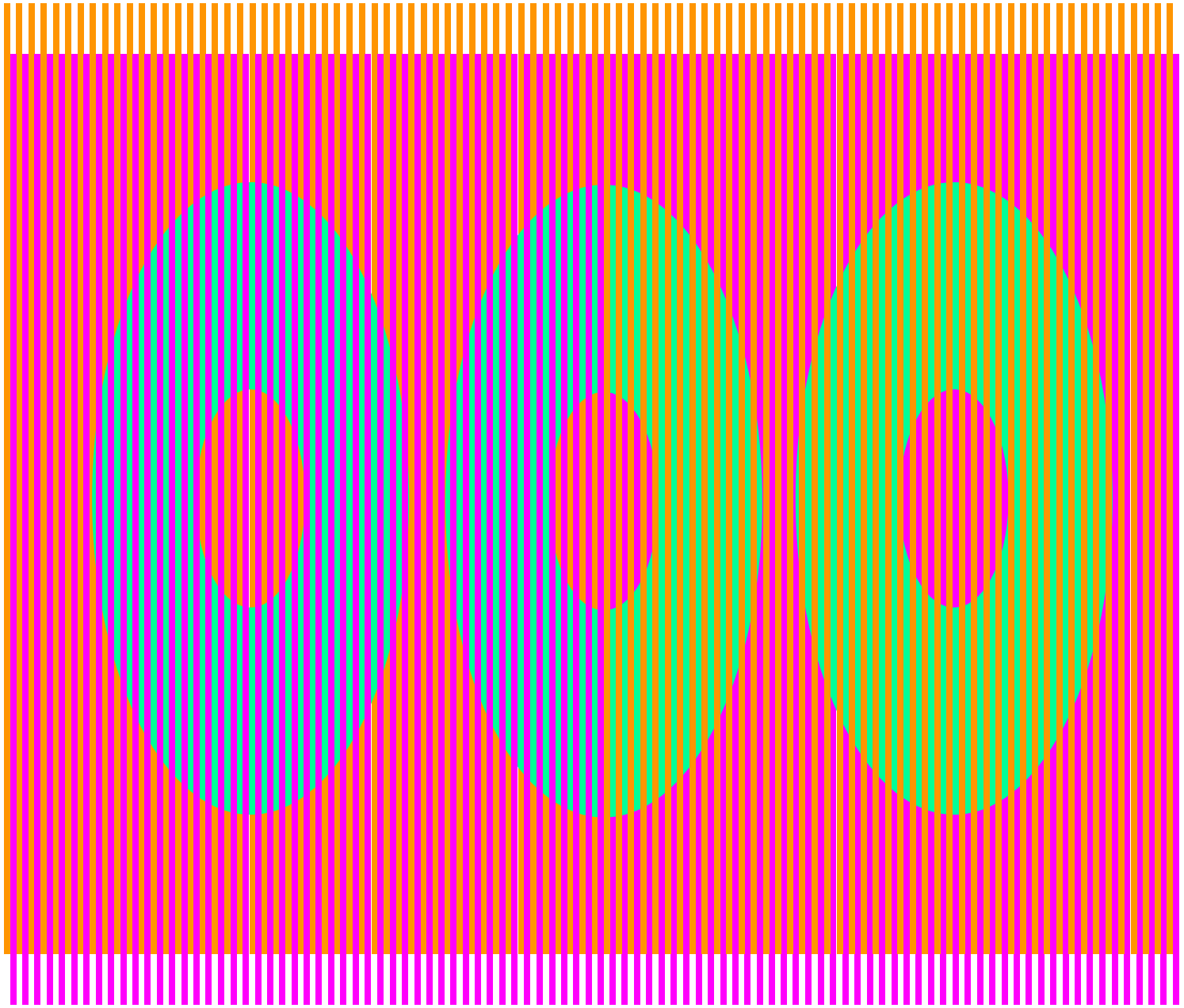




we don't always give our own vision system credit for everything it's doing...



we don't always give our own vision system credit for everything it's doing...



What's red?

Aargh!

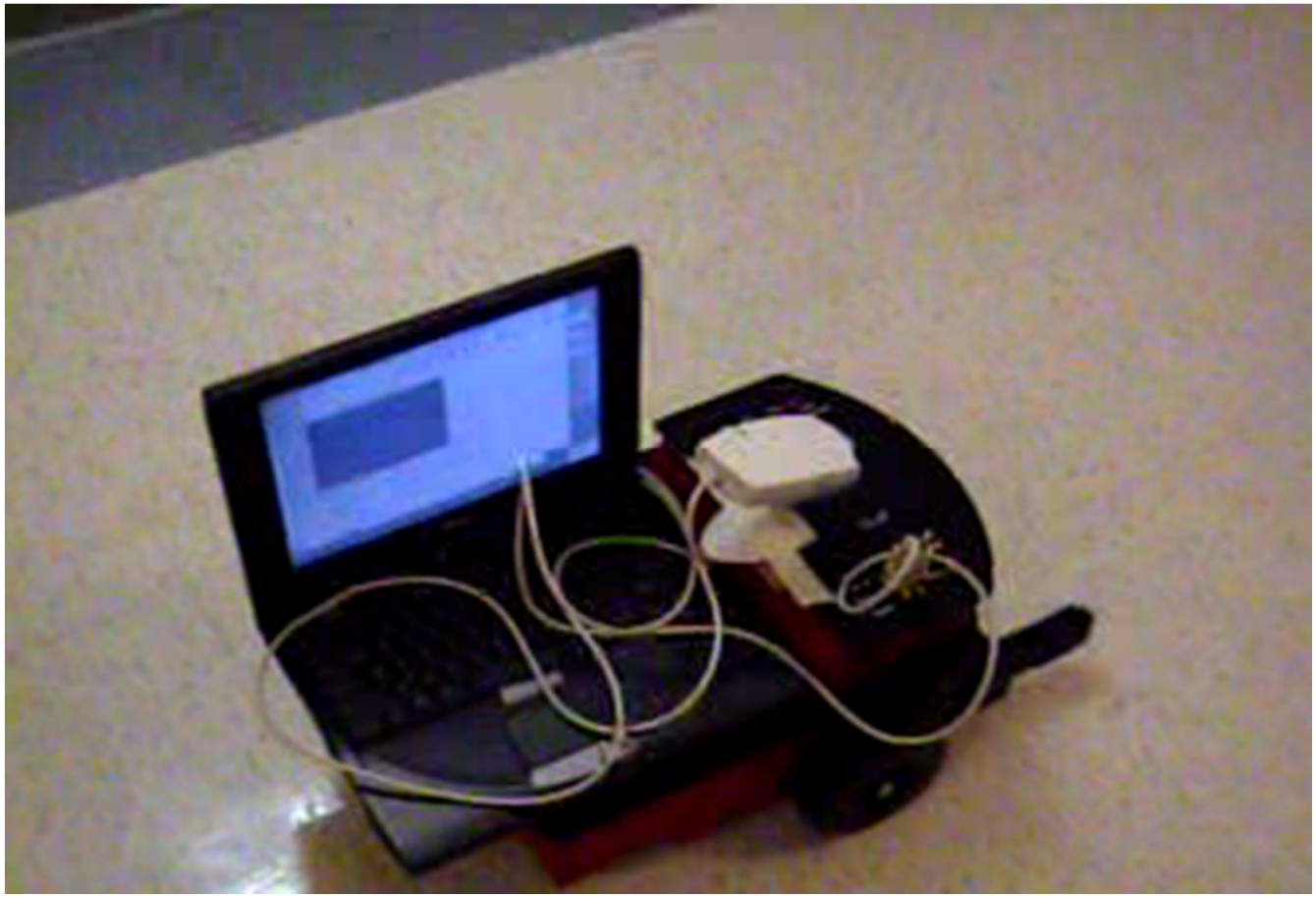
| hue | < 25

saturation > 0.75

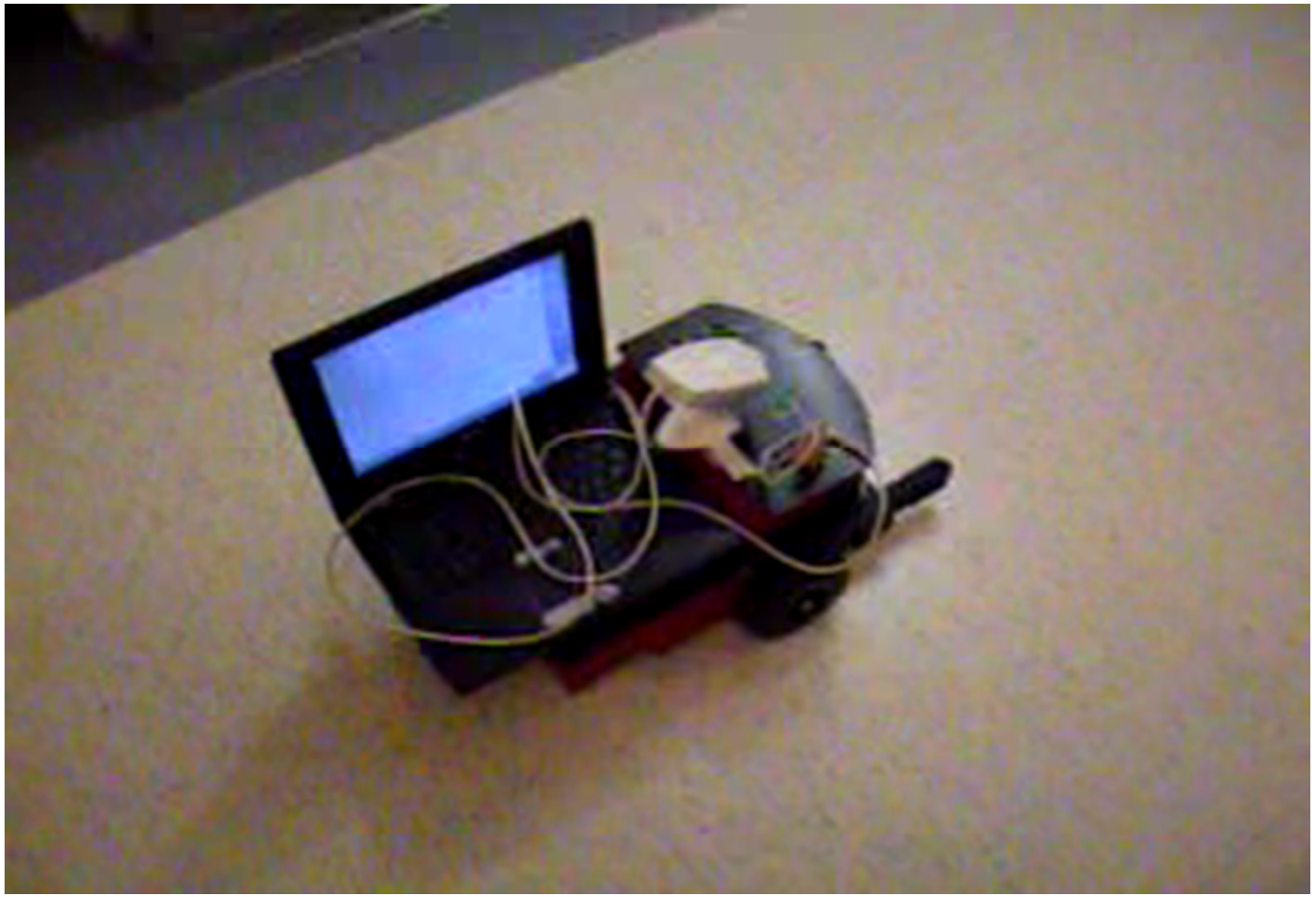
The door is still matched, too… ***why?***

The coke-can collector: ***wall-avoidance***

The coke-can collector: ***seeking...***

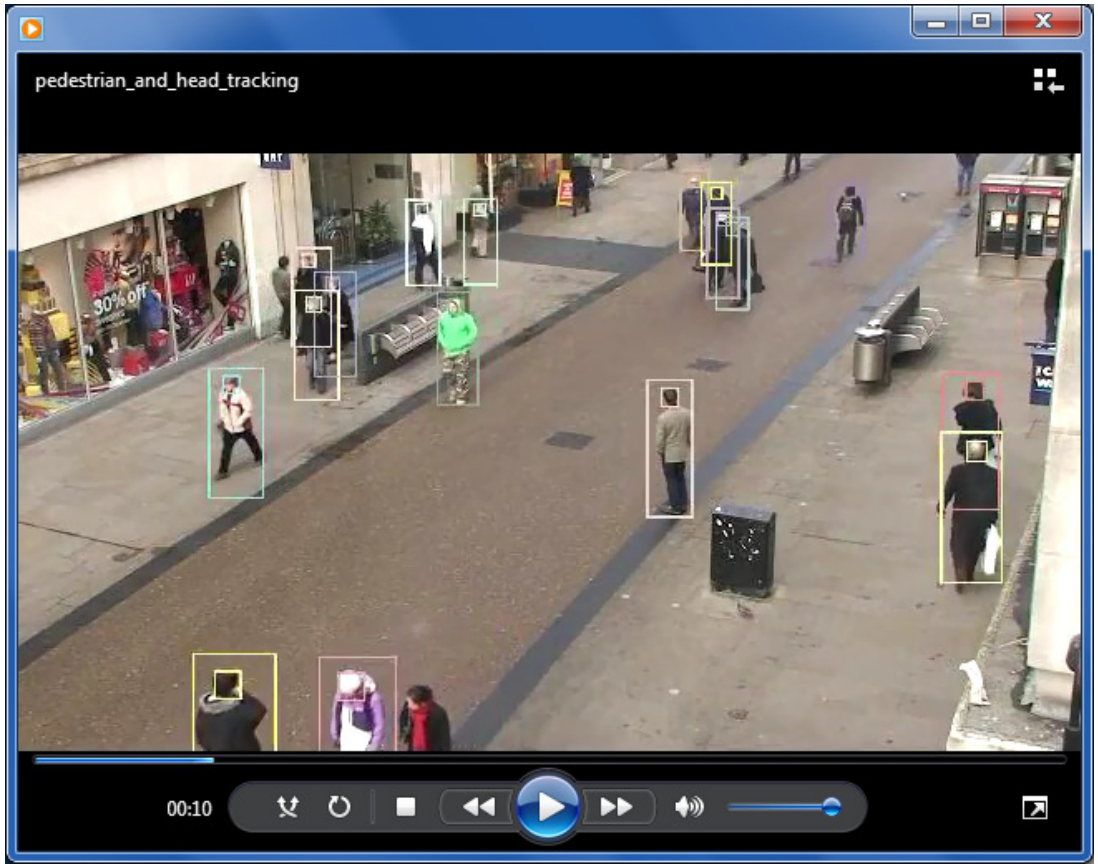


The coke-can collector: ***seeking...***



The coke-can collector: ***success!***

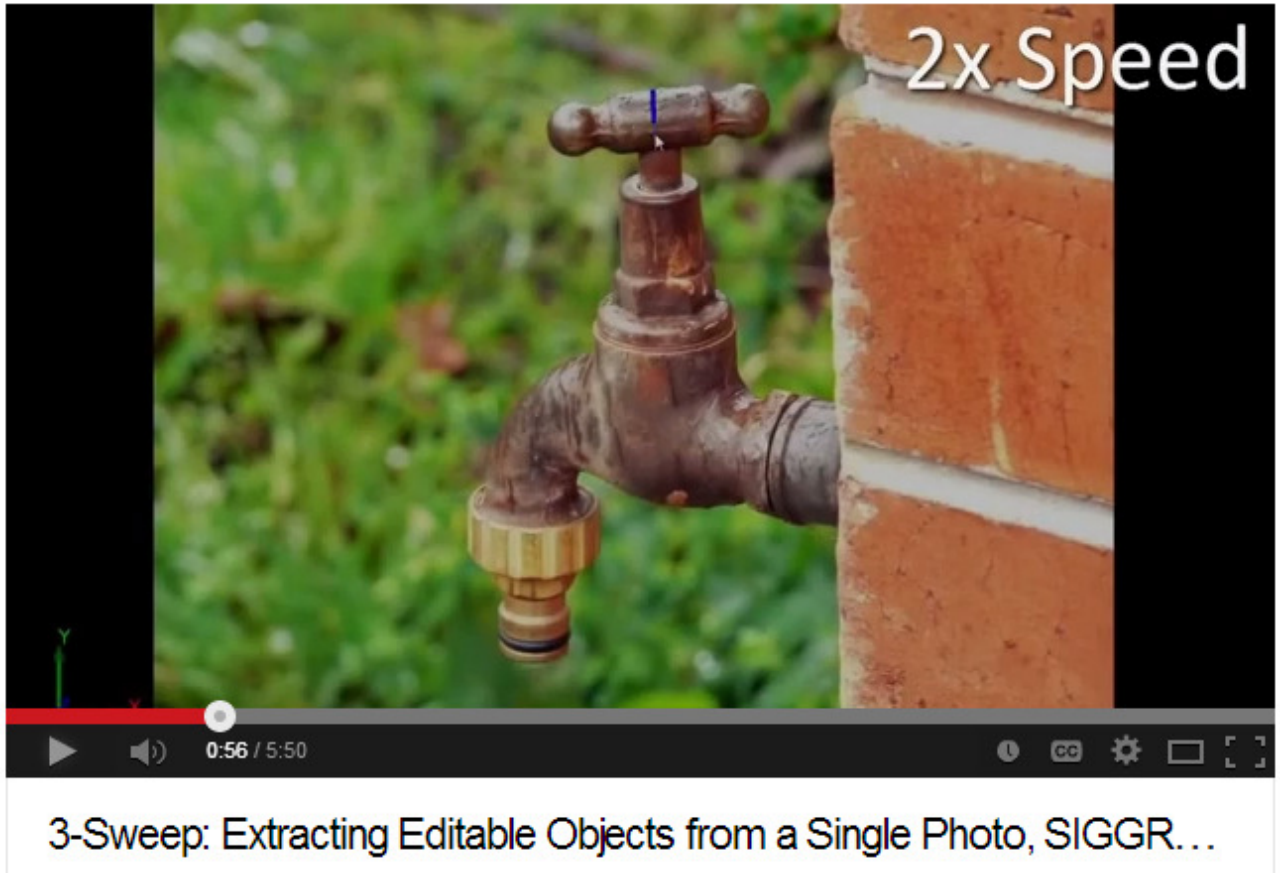
State of the "art" …



Actual output: ***contents***

people, walking…

State of the "art" …

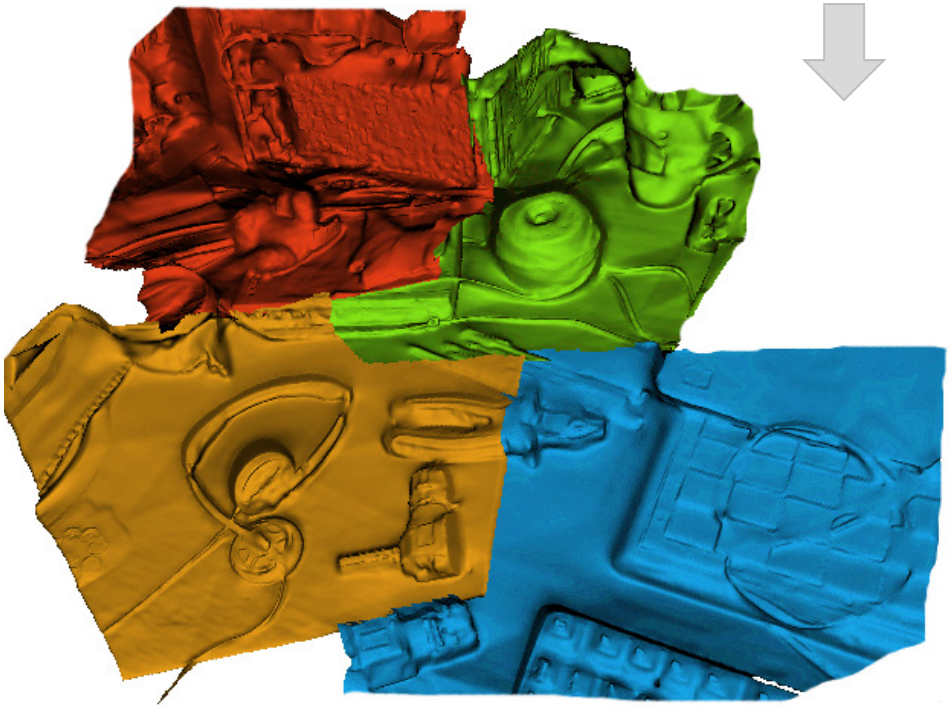


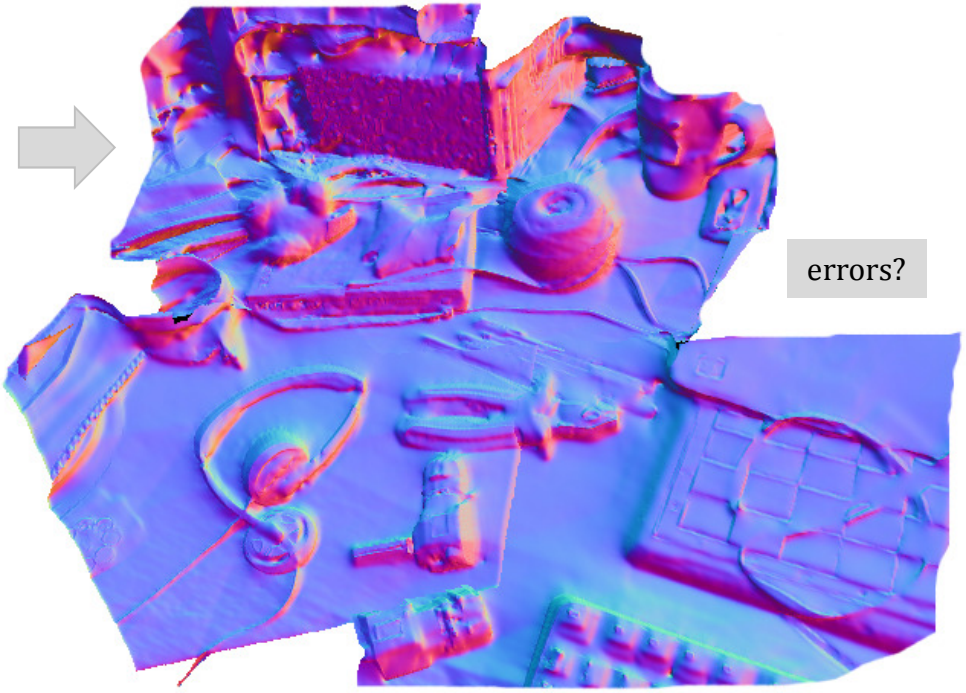
Actual output: ***models***

a tap against a wall…

3d reconstruction ?

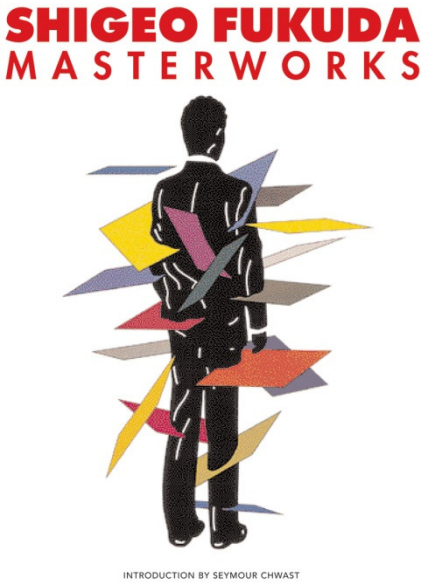
Incredibly, our vision systems can do this.





errors?

An elegant example of the 2d and 3d geometry of vision



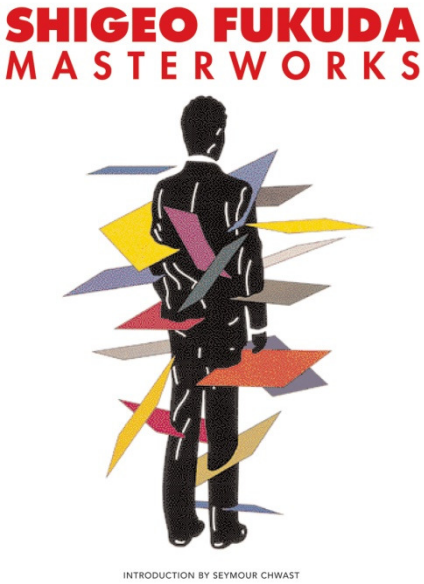
Fukuda's vision atop vision's geometry

Shigeo Fukuda at Color/Value: Icograda Design Week in Daegu, July 2008.

16 January 2009

SHIGEO FUKUDA PASSES AWAY AT 76

Tokyo (Japan) - At the age of 76 years old, Mr. Shigeo Fukuda passed away suddenly in Tokyo on Sunday, 11 January 2009 as the result of a subarachnoid hemorrhage. As one of the founding members and directors of the Japan Graphic Designers Association (JAGDA), Mr. Fukuda contributed profoundly to the development of the Association. The whole creative industry and society of Japan was struck with this sad and sudden news. His funeral has already been held with family members.

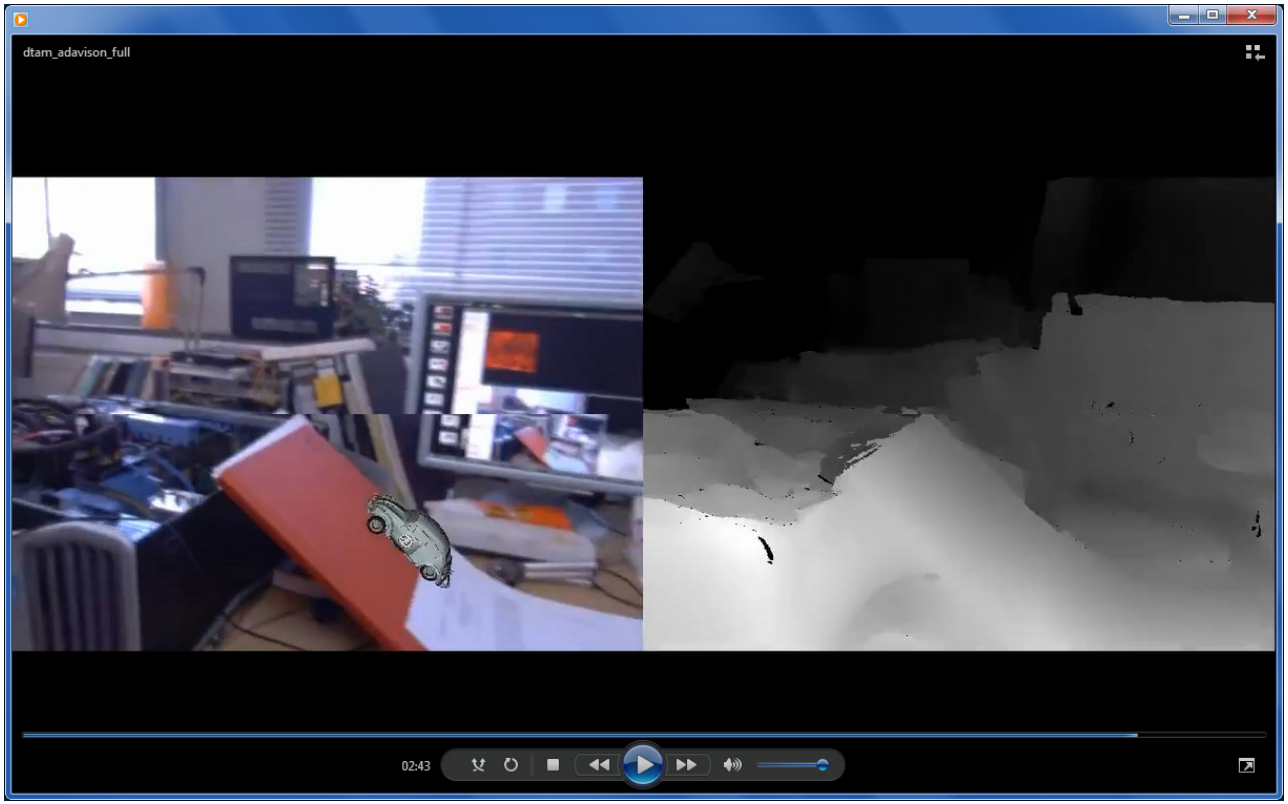


Fukuda's vision atop vision's geometry

3d reconstruction ?

Incredibly, our vision systems can do this.

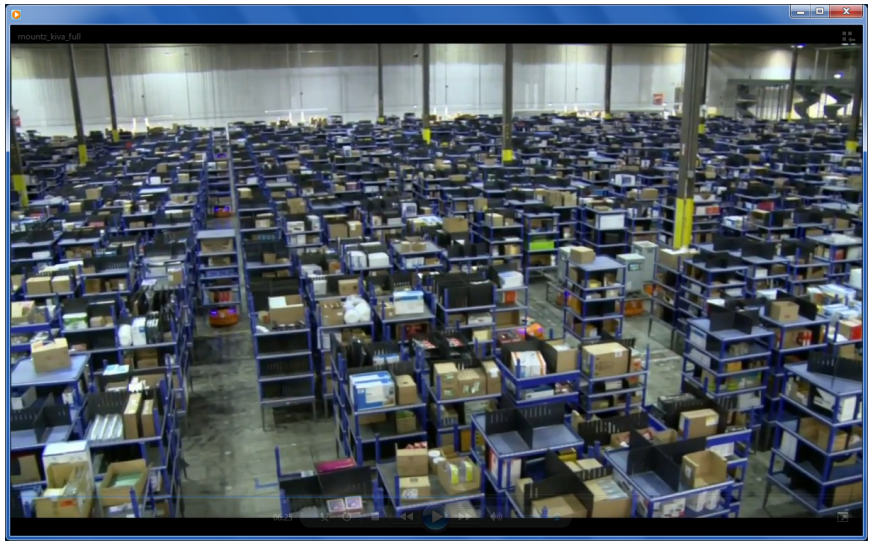
State of the "art" ...

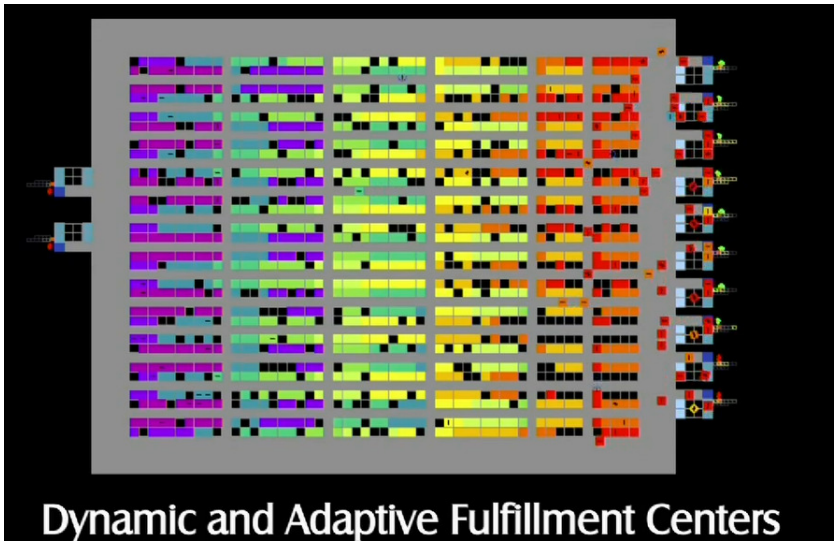


Actual output: ***contents***

3d world model

CS in surprising places…





Automated Material Handling
Order Fulfillment System

Kiva Systems, M. Mountz

CS!

CS 5: the past…

Functions & variables

Recursion

Representations (binary, ascii)

Circuit design & Hmmm

Loops, 2d arrays

Dictionaries

Objects and Classes

Computability

growing trees

looks like broccoli to me!

Caesar cipher

4-bit multiplier

Mandelbrot, Life

Markov Text Gen.

Date, C4, Project

Finite state machines

Uncomputable functions

CS 5: the past…

Functions & variables

Recursion

Representations (binary, as

Circuit design &

Loop

Diction

Objects a Classes

Computability

ing

looks like
broccoli to me!

Future CS?

ultiplier

Mandelbrot, Life

Markov Text Gen.

Date, C4, Project

Finite state machines
Uncomputable functions

Interfaces — *User Interfaces, Graphics, Animation*

CS 124, 155, 157

AI — *AI, Neural Networks, Computer Vision, Robotics*

CS 151, 152, 153, 154

Systems — *Compilers, B… …nguages, Networking, …, Computer Architecture*

Eng 85, 155, 158 & CS 125, 131, 132, 134, 136

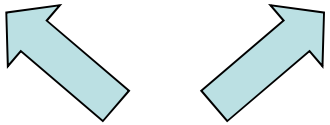
Other CS courses?

Theory — *Theory of Computation, Advanced Algorithms*

Math 167, 168 & CS 141, 142

SW Engineering & *Data structures*

CS 70, 121



Logic & Computability

CS 81

Principles of CS

CS 60

Interfaces

User Interfaces, Graphics, Animation

CS 124, 155, 157

AI

AI, Neural Networks, Computer Vision, Robotics

CS 151, 152, 153, 154

Systems

Compilers, R… …nguages, Networking, …, Computer Architecture

Eng 85, 155, 158 & CS 125, 131, 132, 134, 136

Other CS courses?

Theory

Theory of Computation, Al…

Beyond classes?

SW …

Dat…

CS 70, 12…

Logic & Computability

CS 81

Principles of CS

CS 60

Summer CS

Project pitches: *First Thursday of the spring term…*


Summer CS
Project pitches: First Thursday of the spring term…
you might also consider grading for CS5 next fall – all schools welcome…
but what if I'm nostalgic for CS 5 itself ?

BWAHAHAHAHAHHAHAHHAHAHAH
I AM THE WINNER AND IT ALL WORKS!!!
GWAHAHAHAHAHAHAHAHAH!!

I started at noon, I'm done at 11:11

minus a brief break to watch shawshank redemption and eat dinner,
I've been working solid

I AM TOTALLY THE WINNER!

one of many favorite grading - encountered comments

Final Projects: due Friday @ 5pm

Exam: next Mon. or Wed. at 2 pm (Big Shanahan)

REVIEW: Optional review session covering the practice final and any other questions…

Sunday evening, 7-8+ pm (Big Shanahan)

Final Projects: due Friday @ 5pm

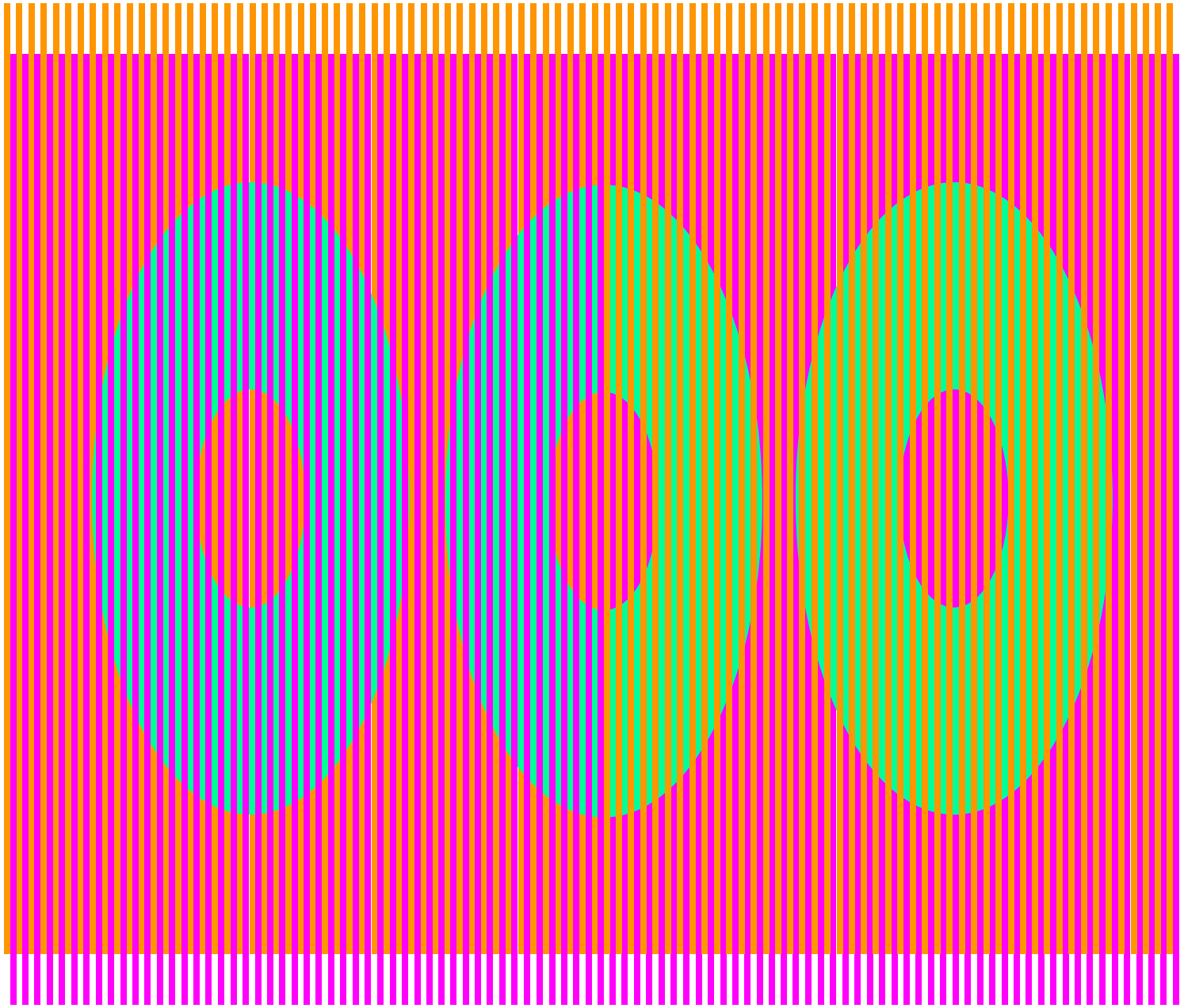
Exam:

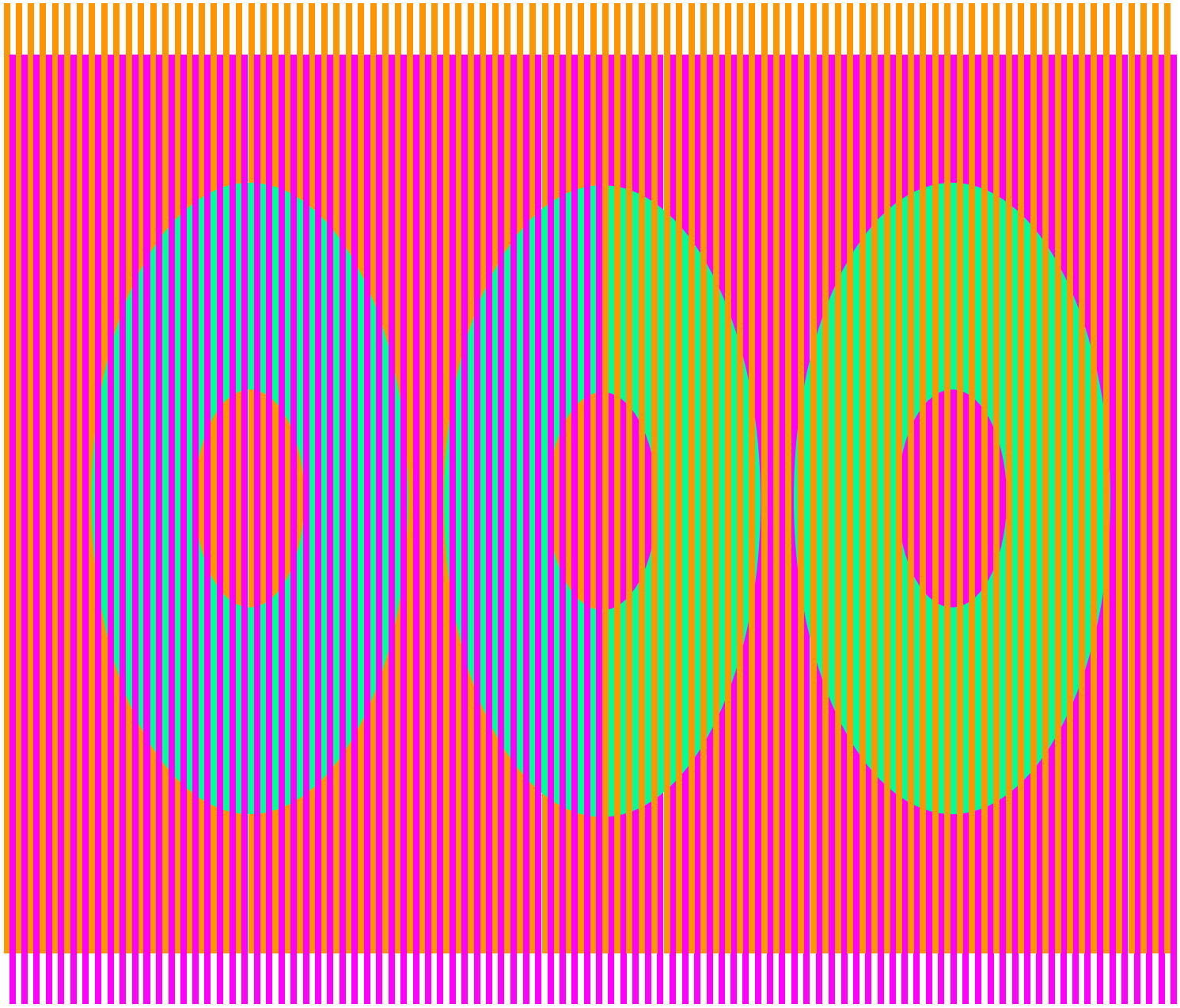
REVIEW

Su **(Big Shanahan)**

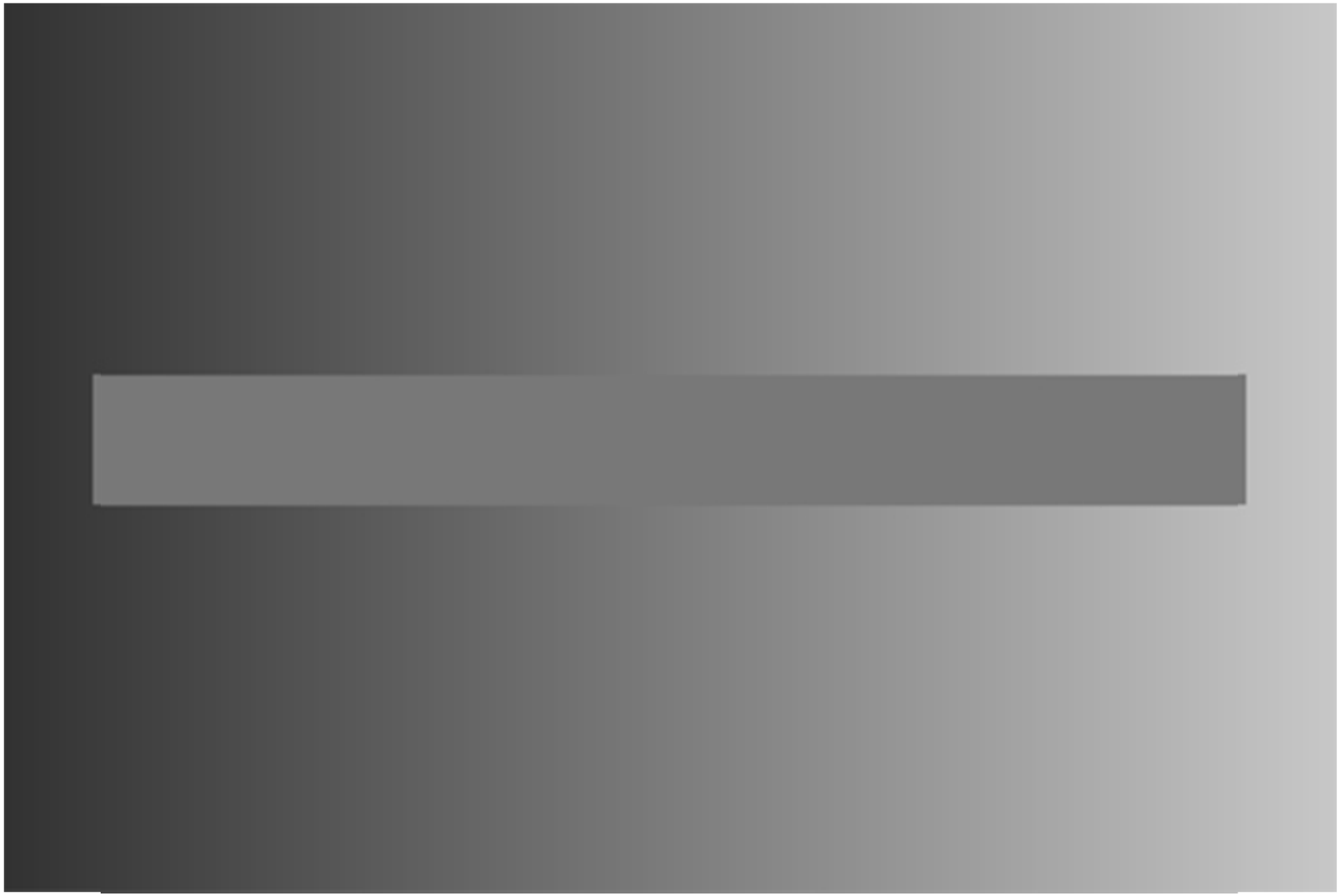
Thank you for joining CS5!

we don't always give our own vision system credit for everything it's doing...

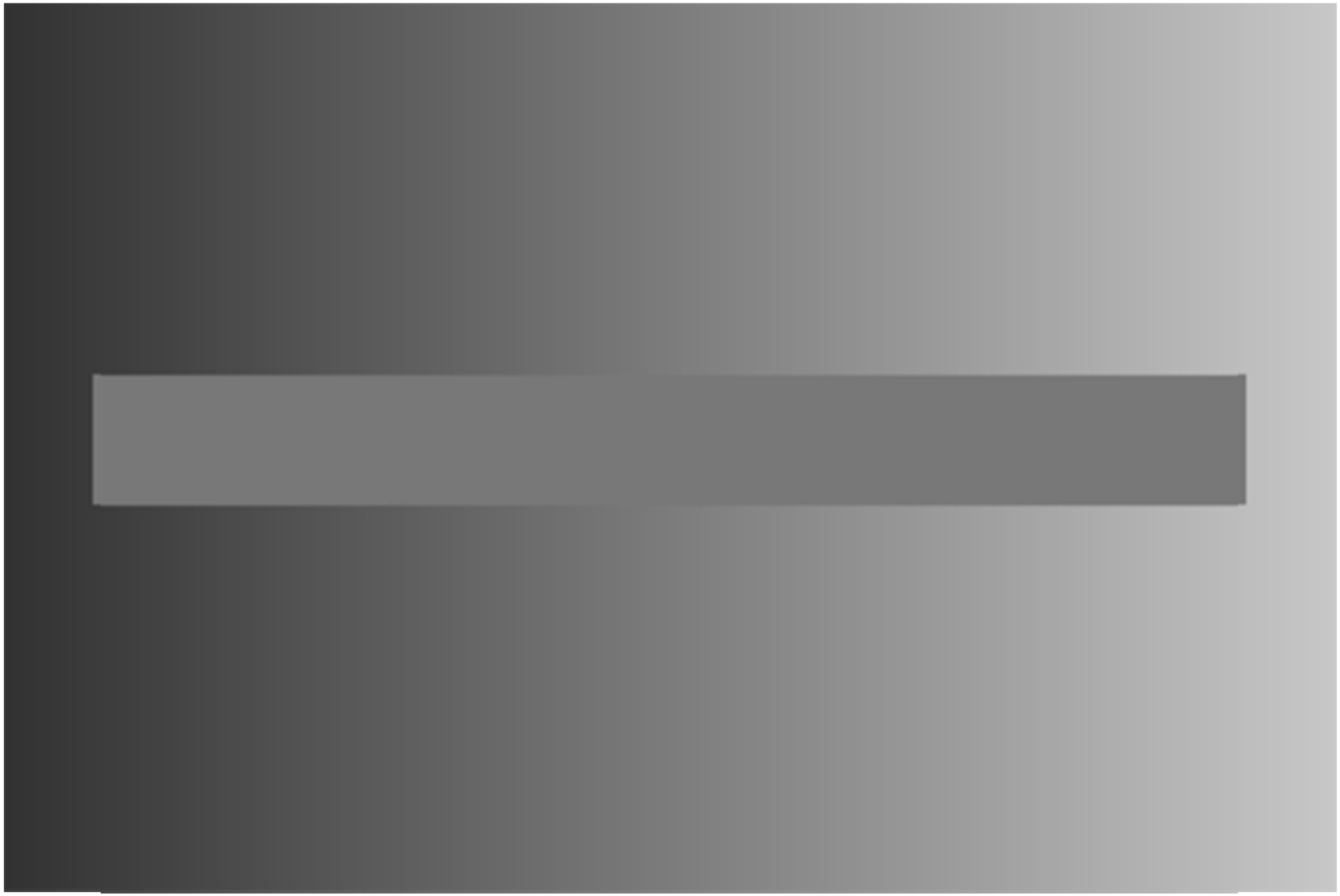




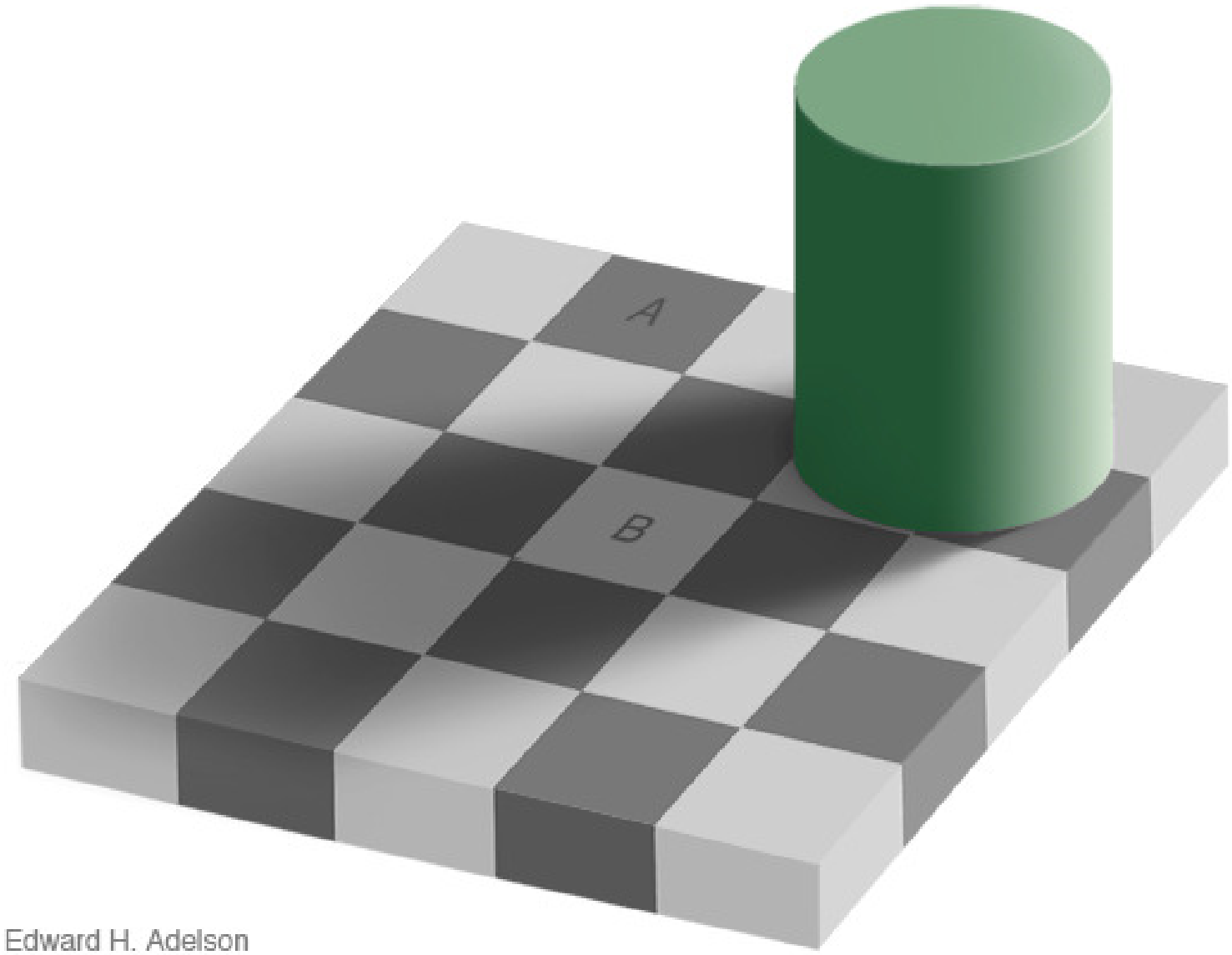
human vision can be deceived, too!



2 layers

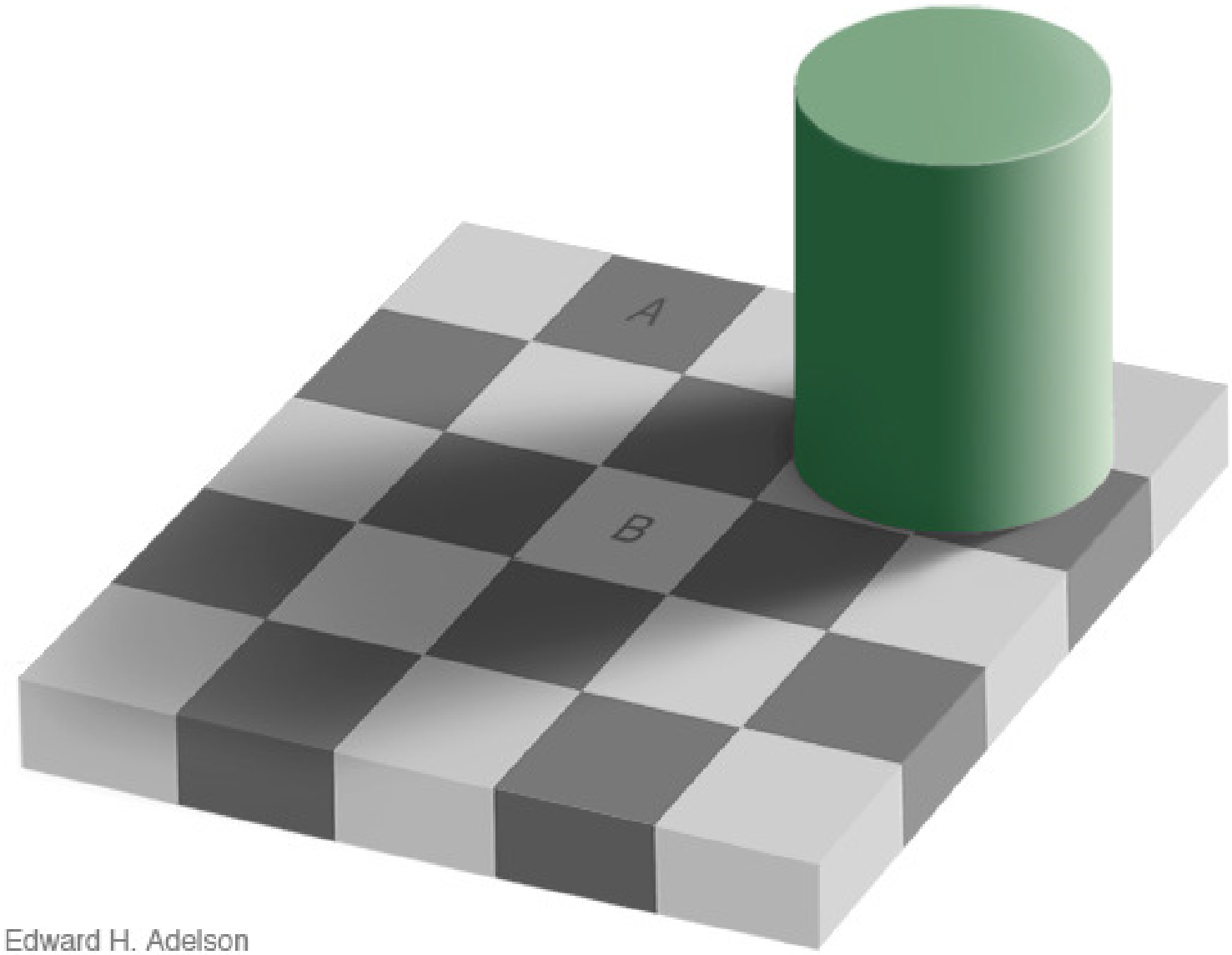


2 layers



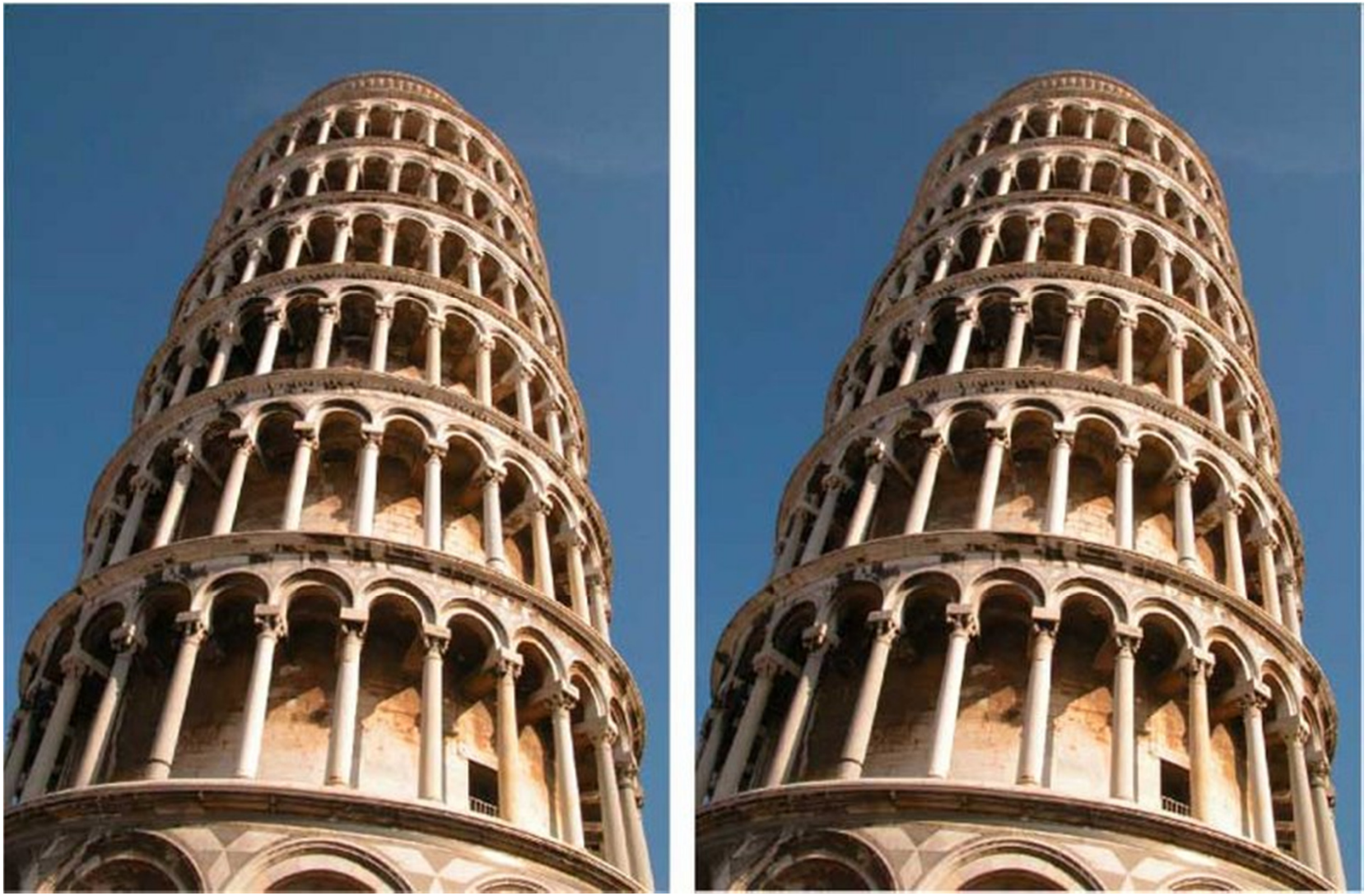
Edward H. Adelson

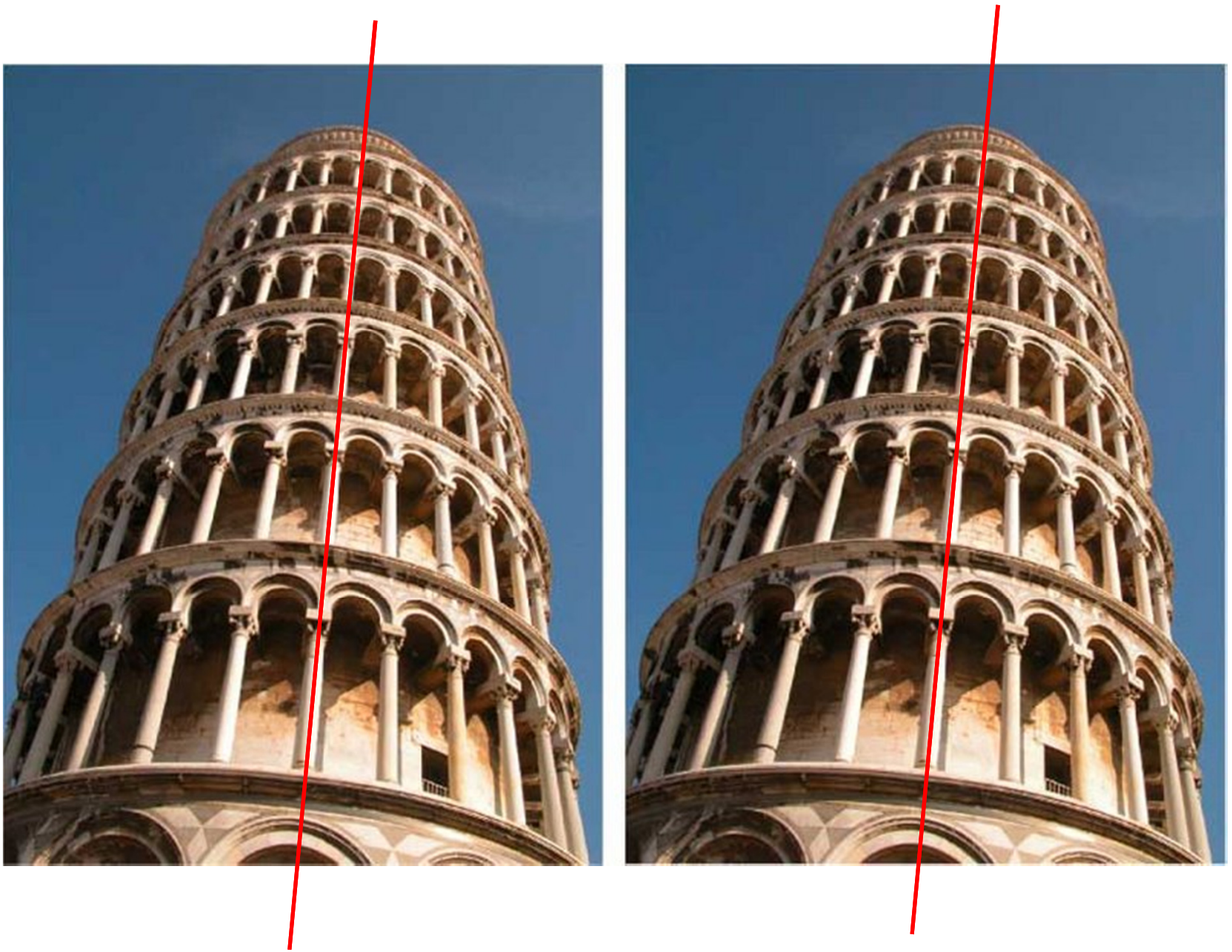
2 layers

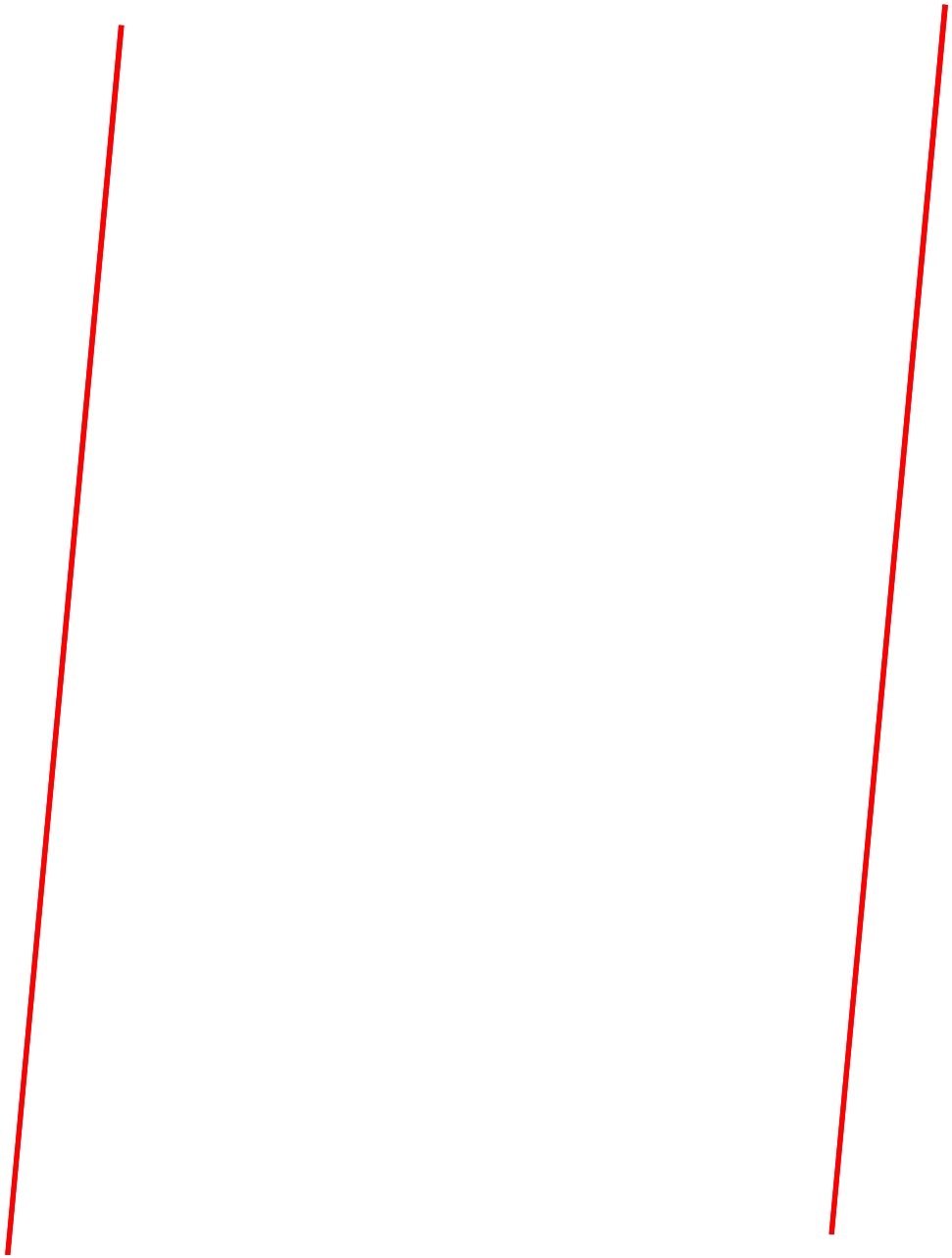


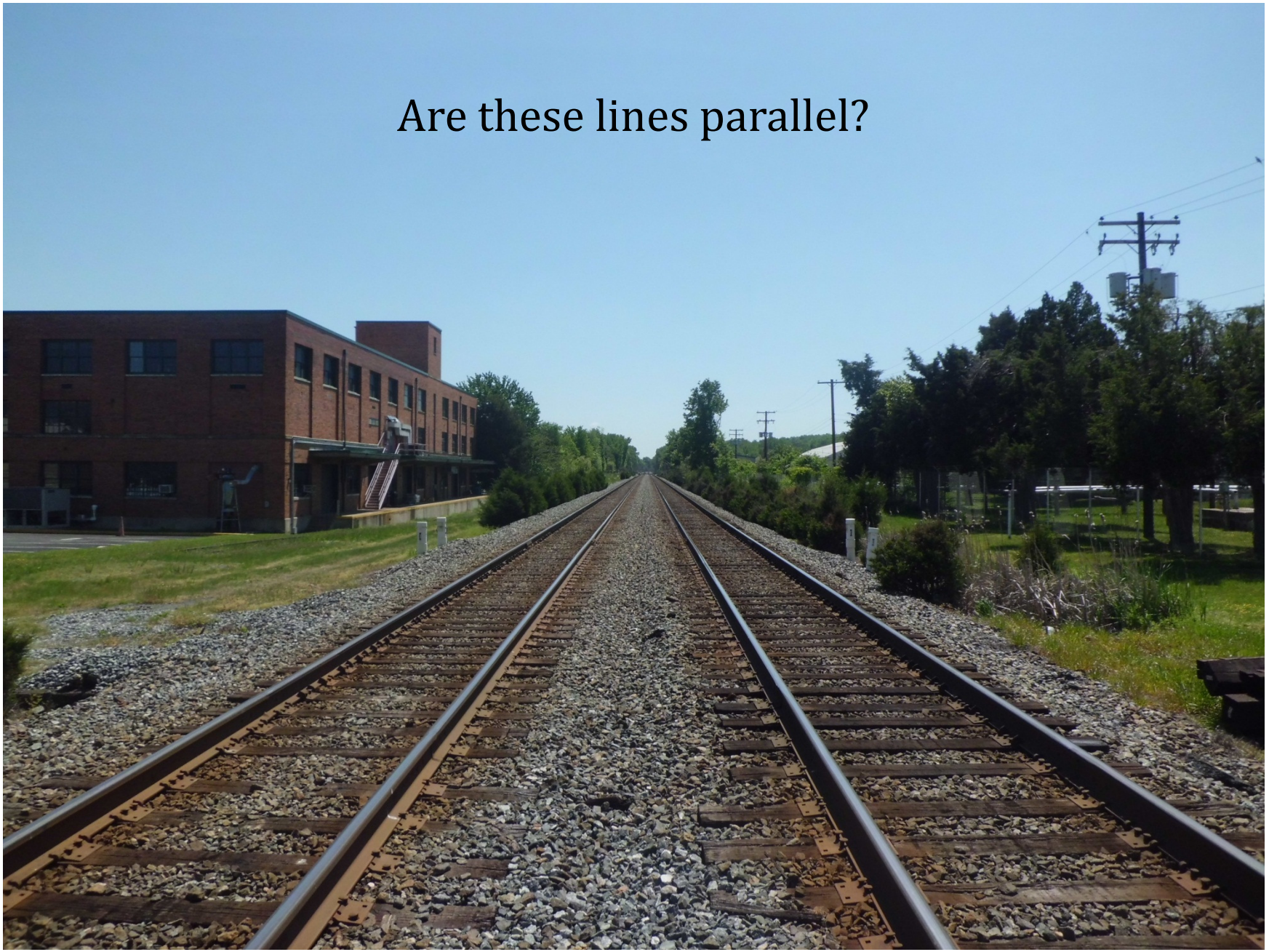
Edward H. Adelson

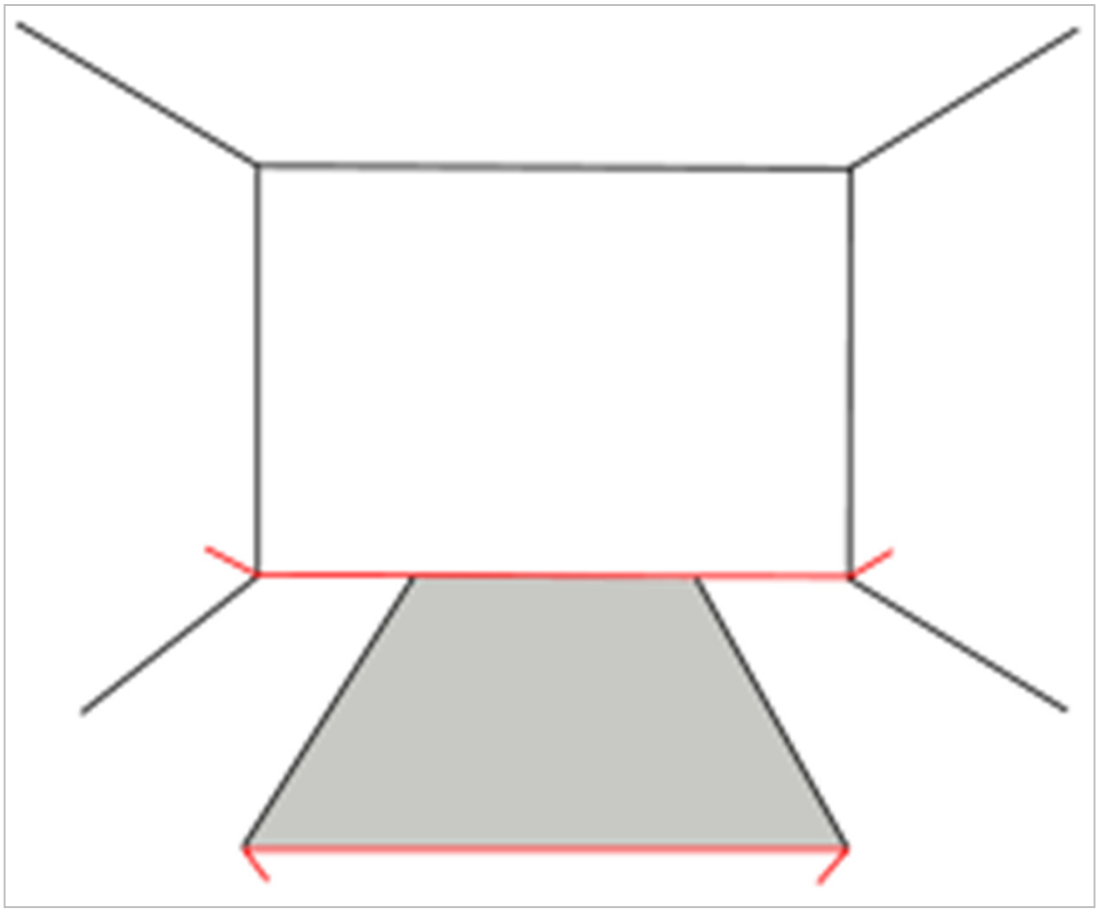
2 layers

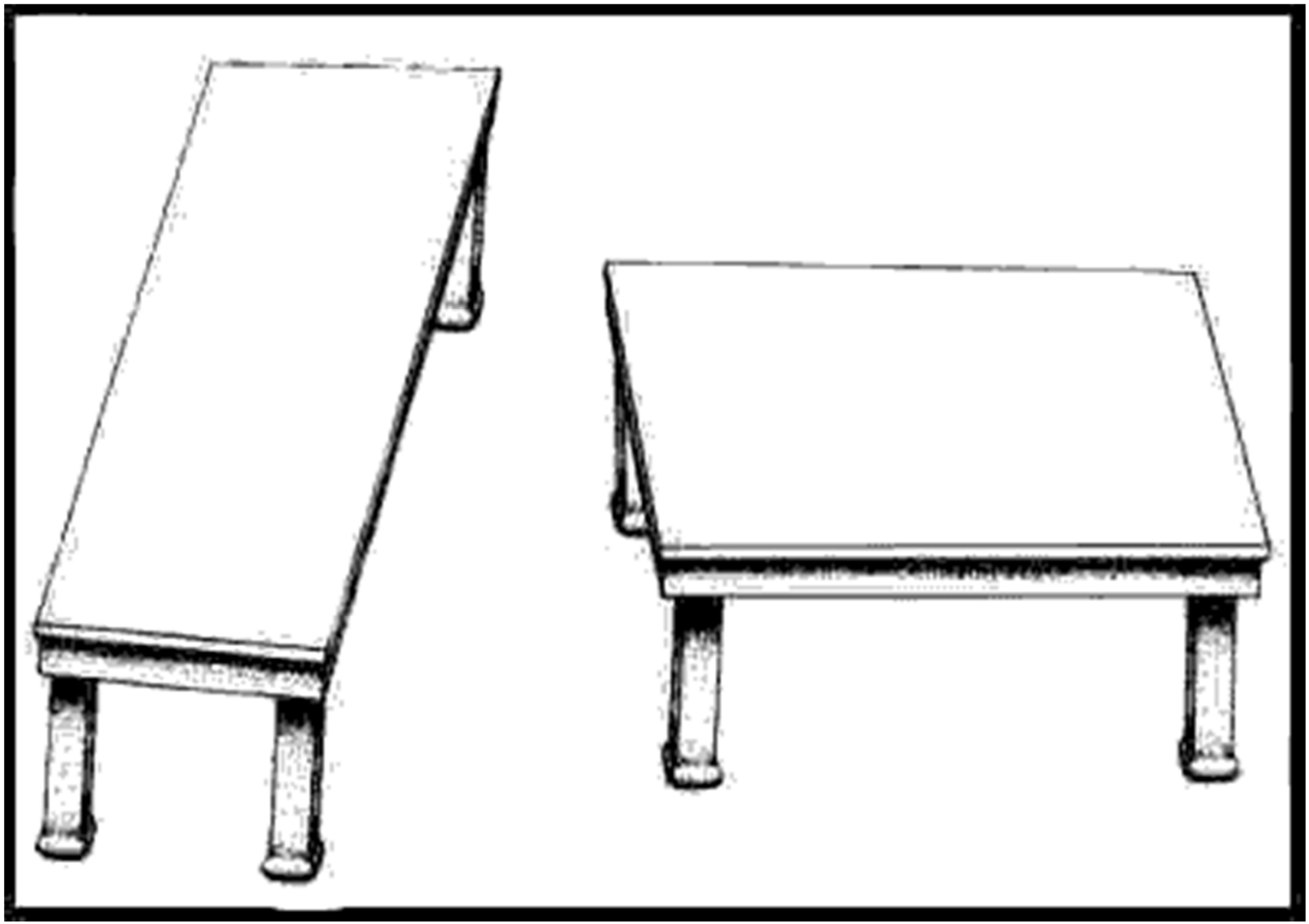


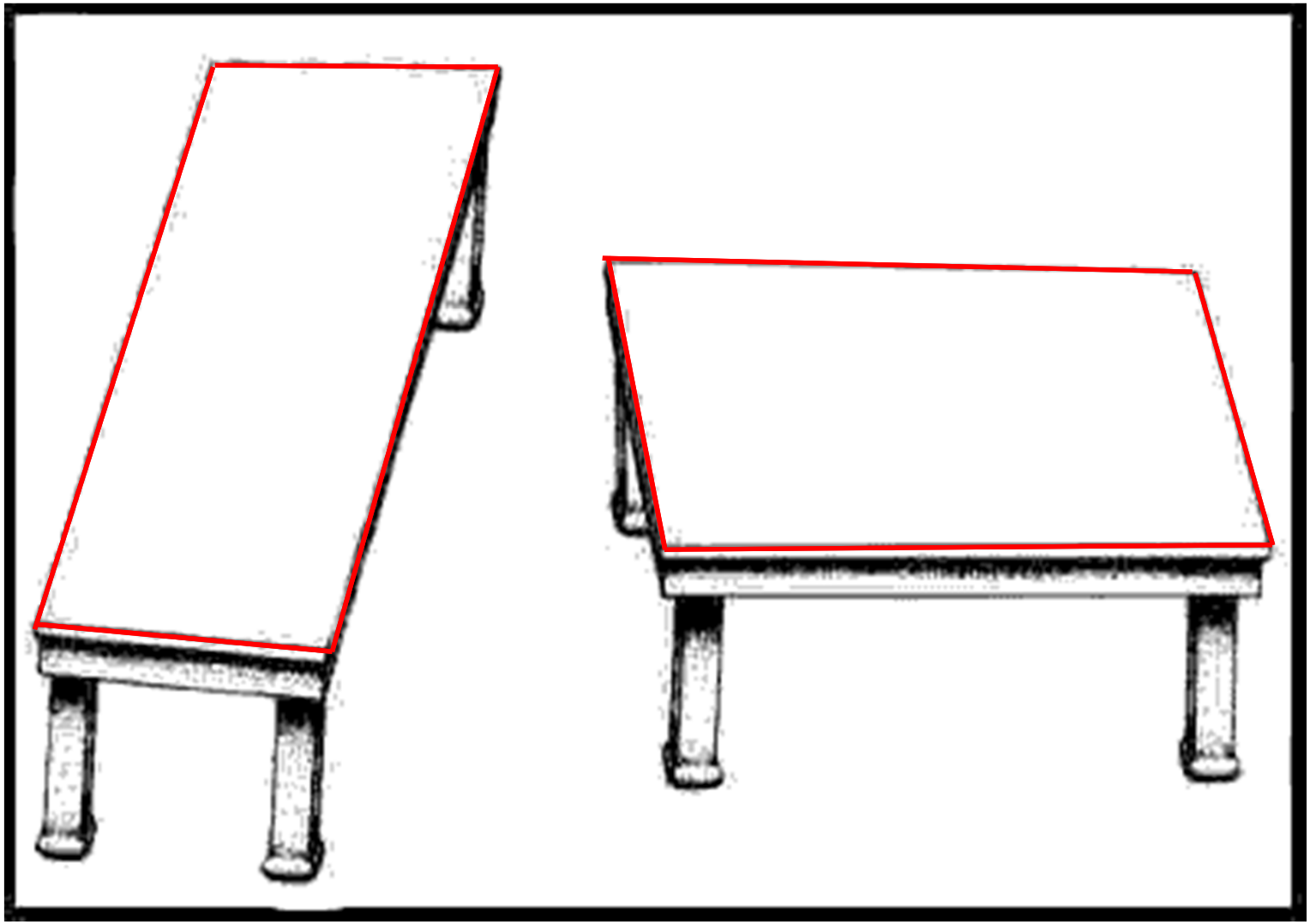


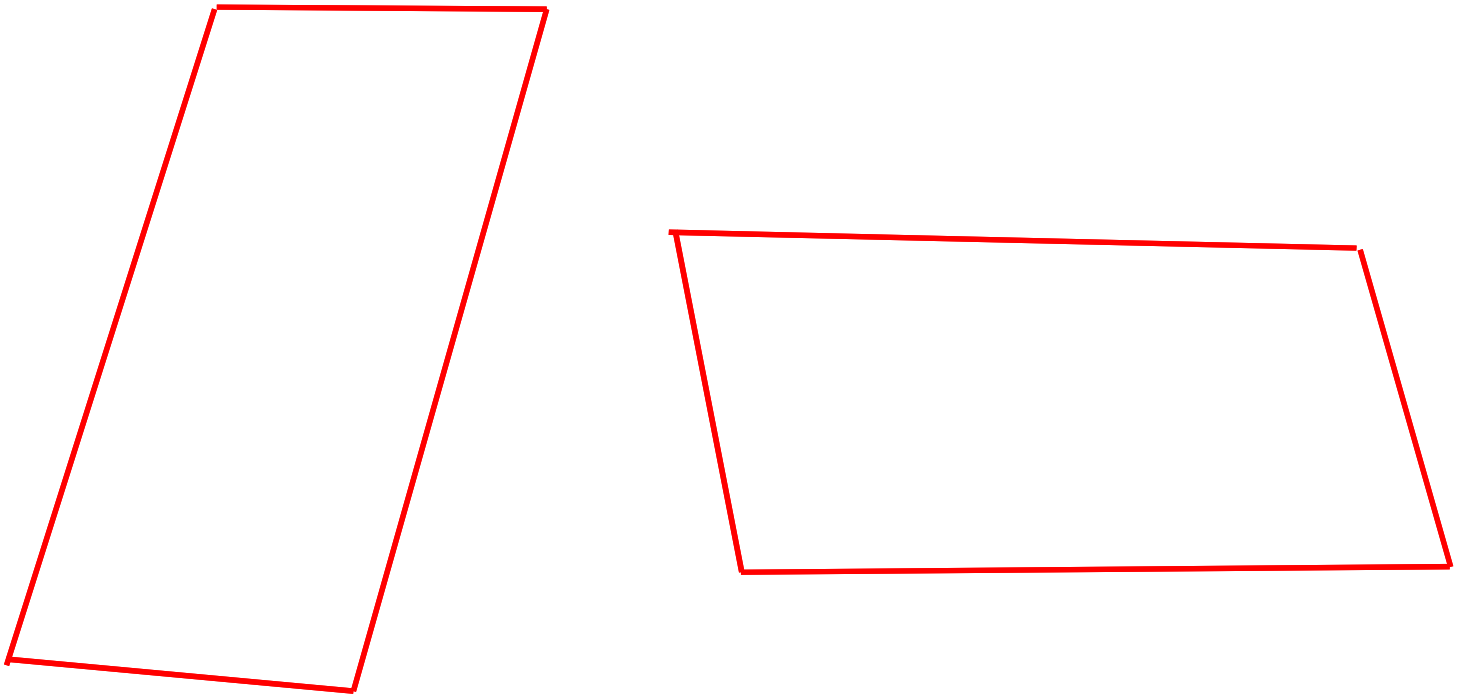



Are these lines parallel?





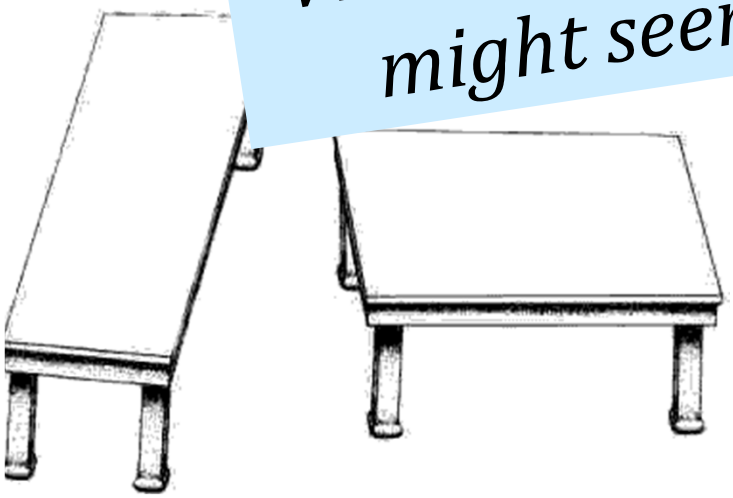


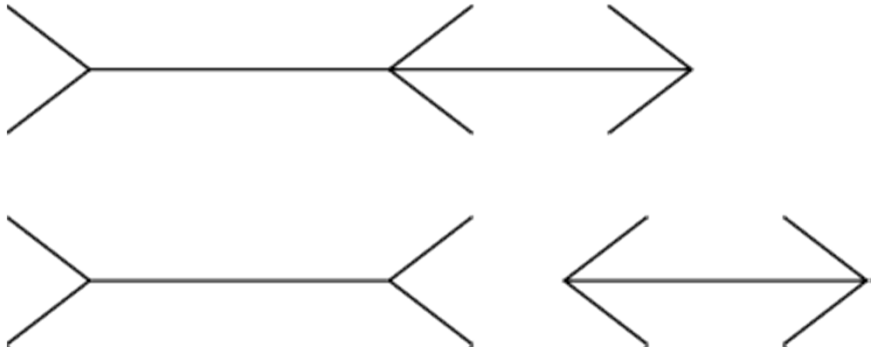


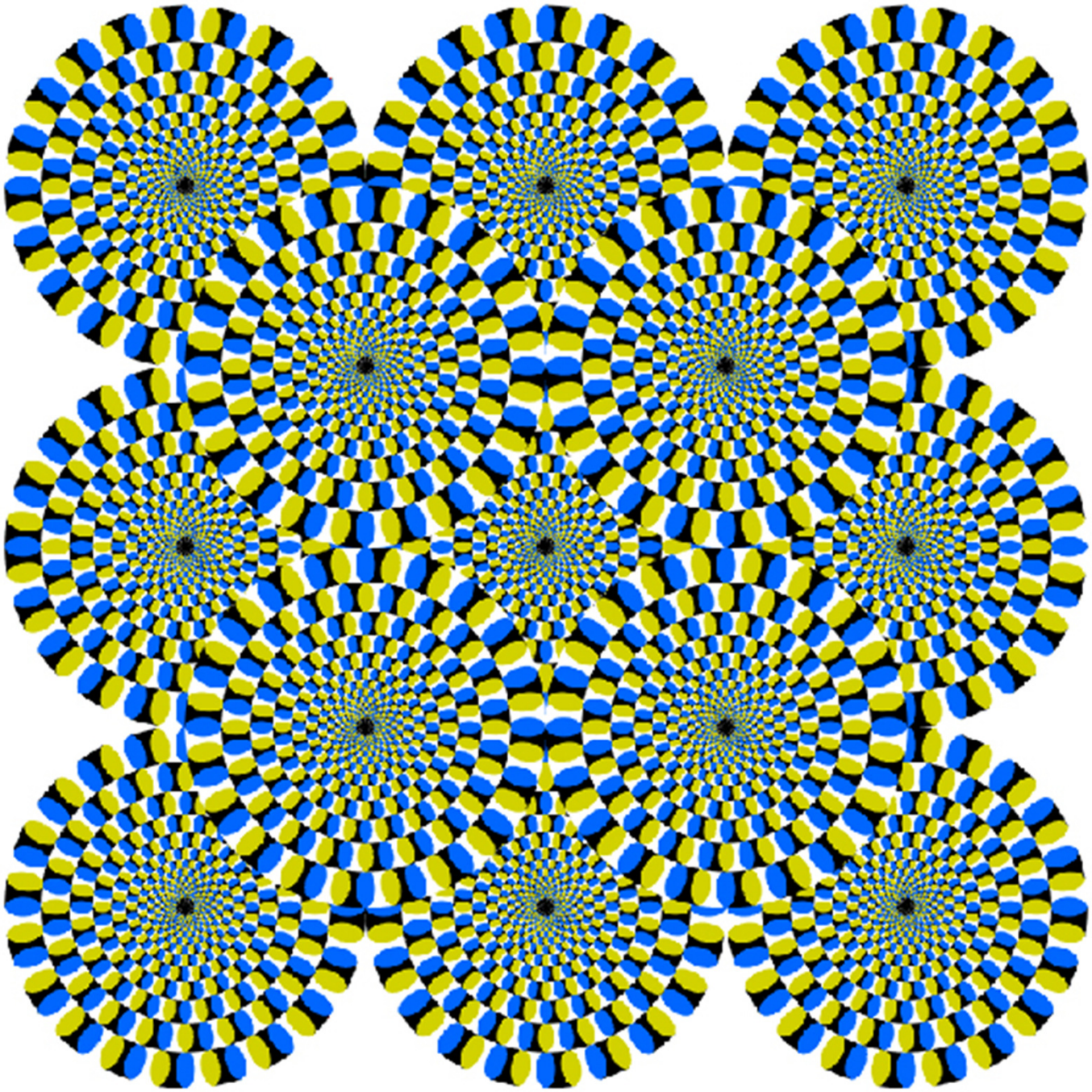
Quiz

Illusions? What computations is your brain doing to cause them?

Vision is more challenging than it might seem on first "glance"!







Interfaces

User Interfaces, Graphics, Animation

CS 124, 155, 157

AI

AI, Neural Networks, Computer Vision, Robotics

CS 151, 152, 153, 154

Systems

Compilers, Programming Languages, Networking, Operating Systems, Computer Architecture

Eng 85, 155, 158 & CS 125, 131, 132, 134, 136

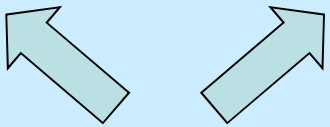
Theory

Theory of Computation, Advanced Algorithms

Math 167, 168 & CS 141, 142

SW Engineering &
Data structures

CS 70, 121



Logic & Computability

CS 81

Principles of CS

CS 60

CS60 ~ Other *Languages*

Racket

```
(define (whoami n)
    (if (= n 0)
      1
      (* n (whoami (- n 1)))))
```

Java

```
public static int whoami(int n)
{
    if (n<2) return 1;
    else return n* whoami(n-1);
}
```

Prolog

```
whoami(0,1).
whoami(N,F) :-  N1 is N-1,
        whoami(N1,F1), F is N*F1.
```

Who is **whoami** ?

Algorithms & efficiency

Suppose your list has **N** elements…

```
L = [4, 7, 42, 1, …, 0, 14, -5, 5]
```

Find `L`'s max

Sort `L`

Will sorting or max-finding take more work?

Algorithms & efficiency

Suppose your list has **N** elements…

```
L = [4, 7, 42, 1, …, 0, 14, -5, 5]
```

Find `L`'s max ? ⇦ Find `L`'s median ? ⇨ Sort `L`

How much work will median-finding take?

Software Engineering

CS 121

'chunky'

Data structures

CS 70

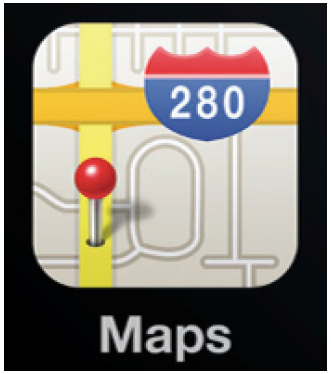
Principles of CS

CS 60

Systems and Software

Corporate Address

Apple
1 Infinite Loop
Cupertino, CA 95014
408.996.1010



the iPhone's icon for Google Maps …

Future CS ?

but what if I'm nostalgic for CS 5 itself ?

CS's Summer projects colloquium

you might consider **grading** for CS5 next fall – all schools welcome...

Most likely:

Th., January 19, 2012
~ place TBA

(probably right here)

CS 5 finale

I'll be back...

CS 5 Practice Exam

CS 5 Review Session

4-5pm Sunday, 12/16 here in Pryne

CS 5 Final Exam (either day is OK!)

2pm - on Thu. or Fri. 12/17-18 here in Pryne

2 pages of notes are OK

practice exam is a good review

CS 5 Final Projects

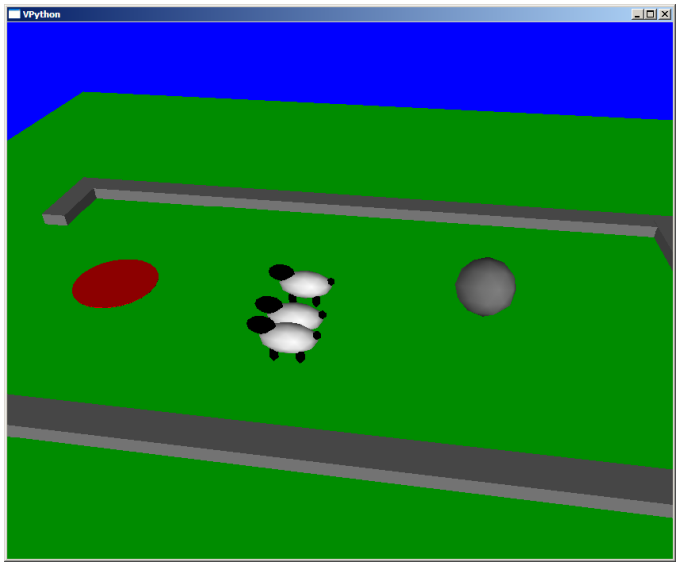
due Fri. 11:59 pm (or Sun. w/ Euro)

Looking back! evals, ideas

Looking ahead? Other options

That's my line!

Rare photograph of CA governor before his morning make-up session.



Most creative pocket design...

CS in the next few years...

the long-awaited emergence of Artificial Intelligence?

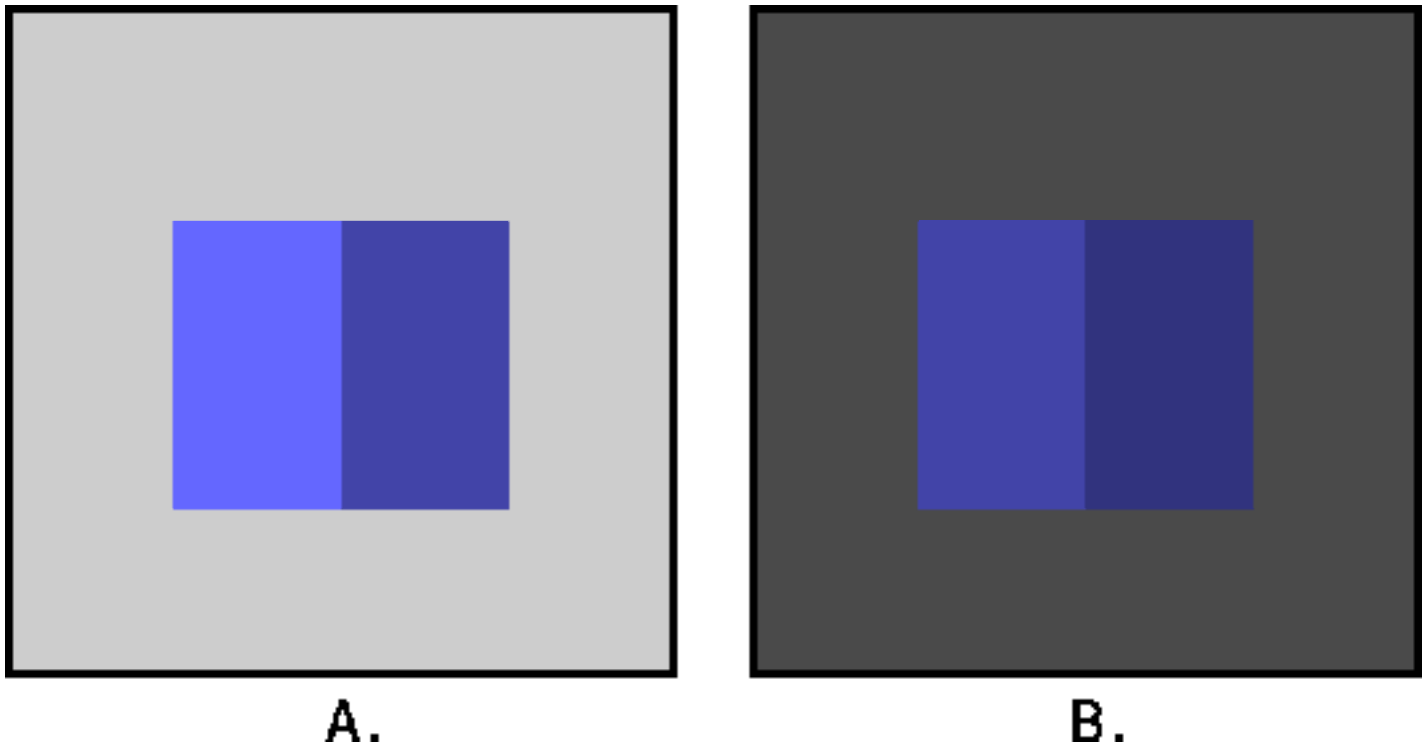
Natural speech interactions — Siri+ ?

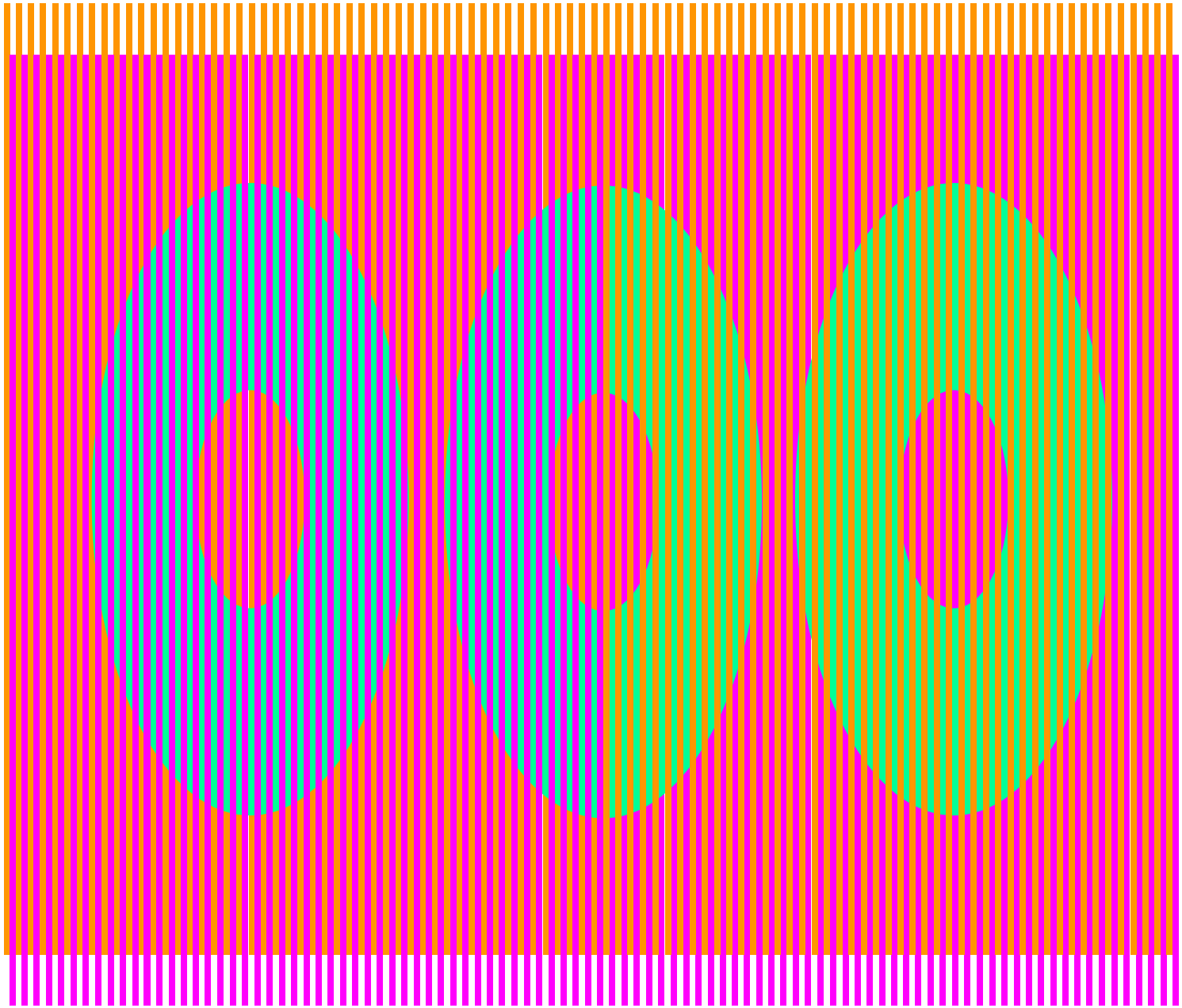
Interpretation of images — Google + facebook recognition

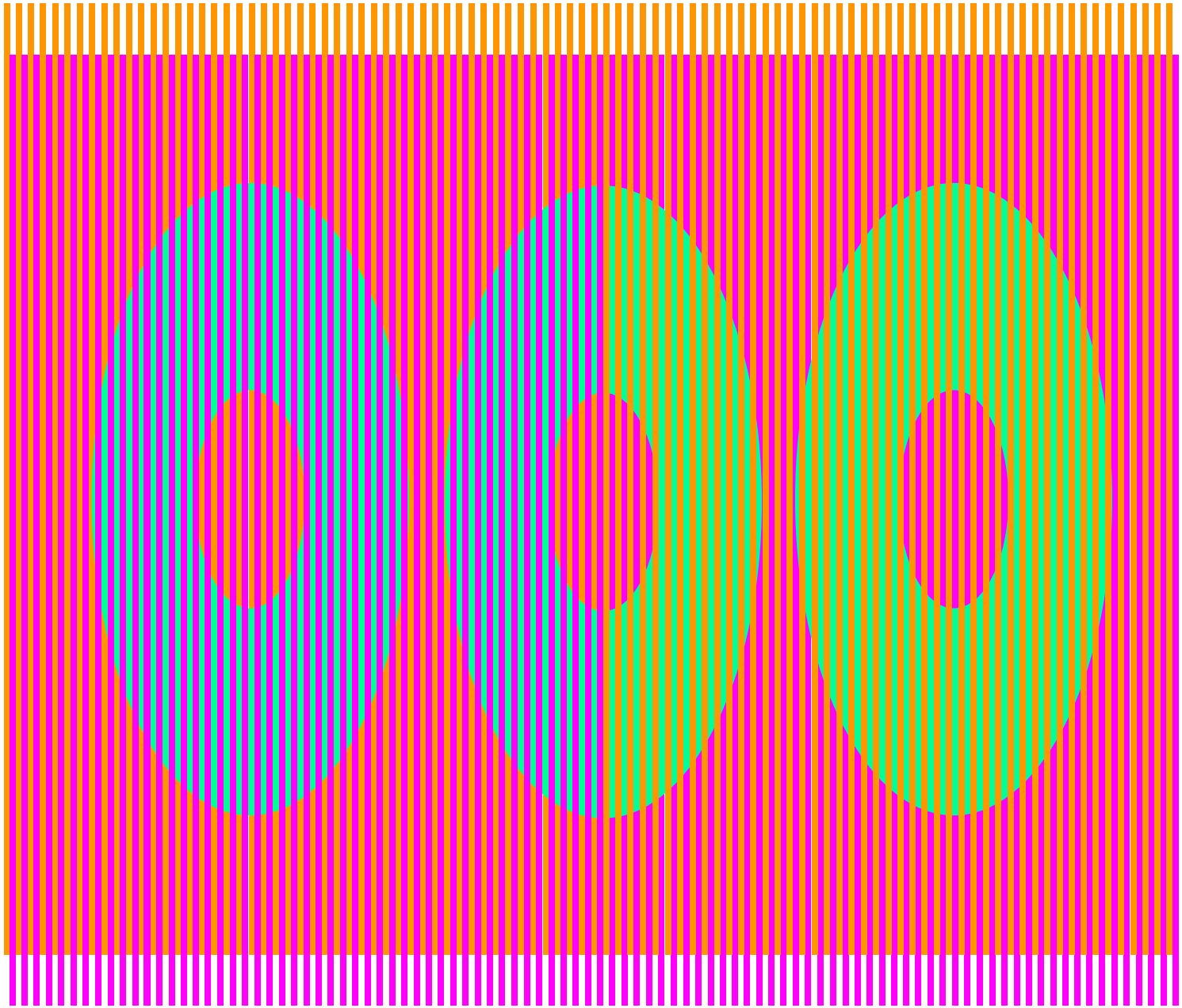
Physical interactions with the world... — autonomous cars?

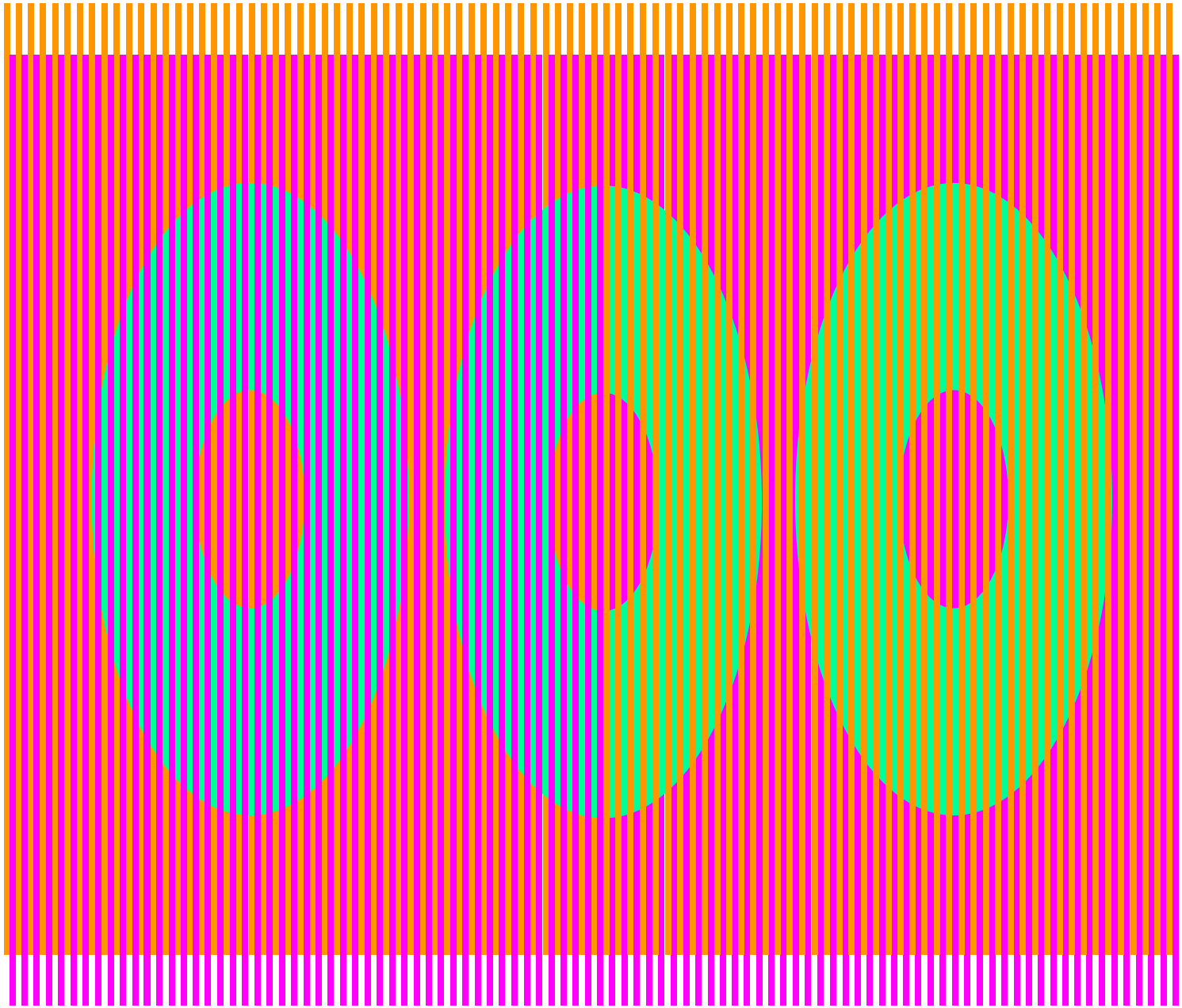
Color Tracking

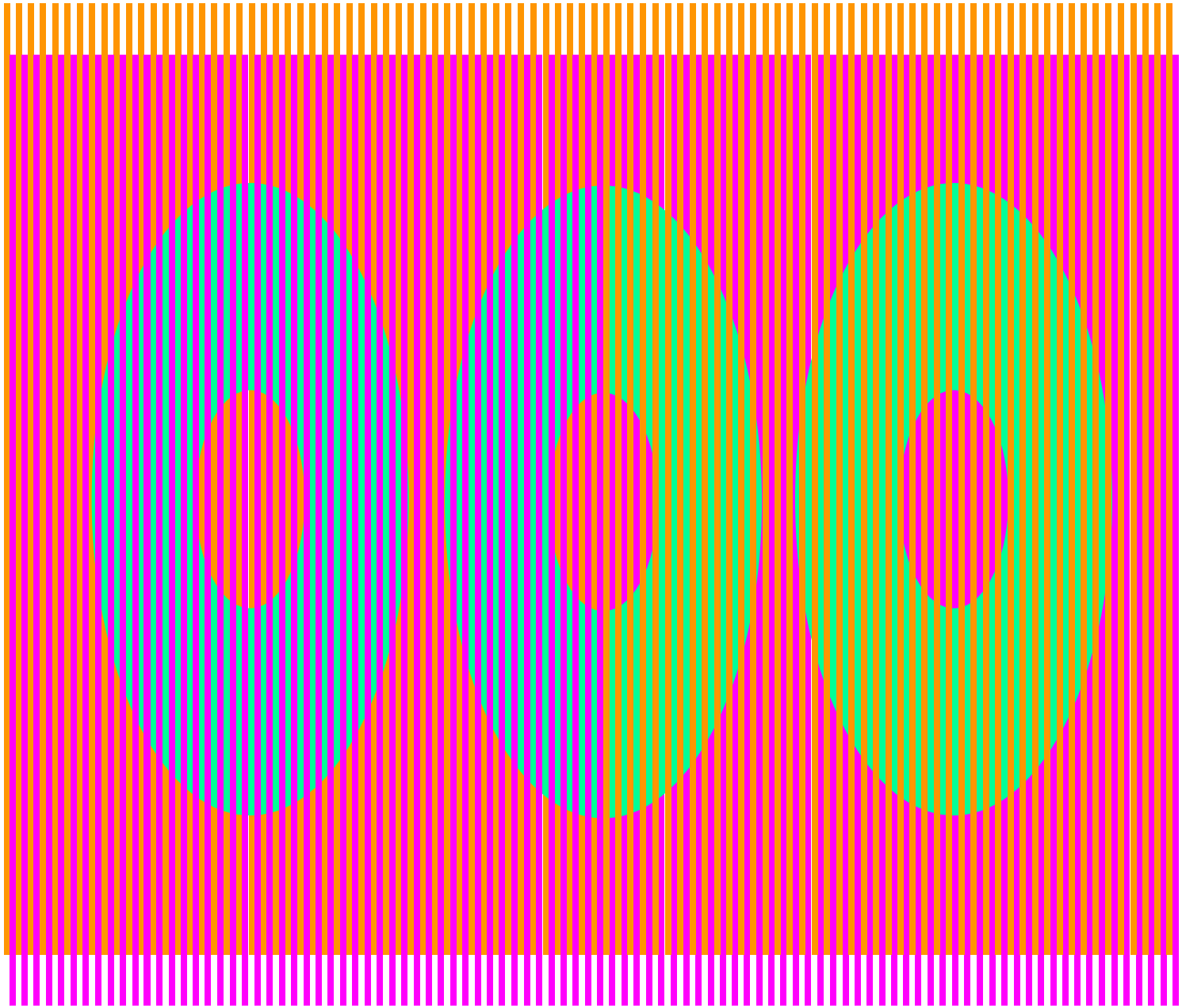
relying on our perceptions of color can be deceiving

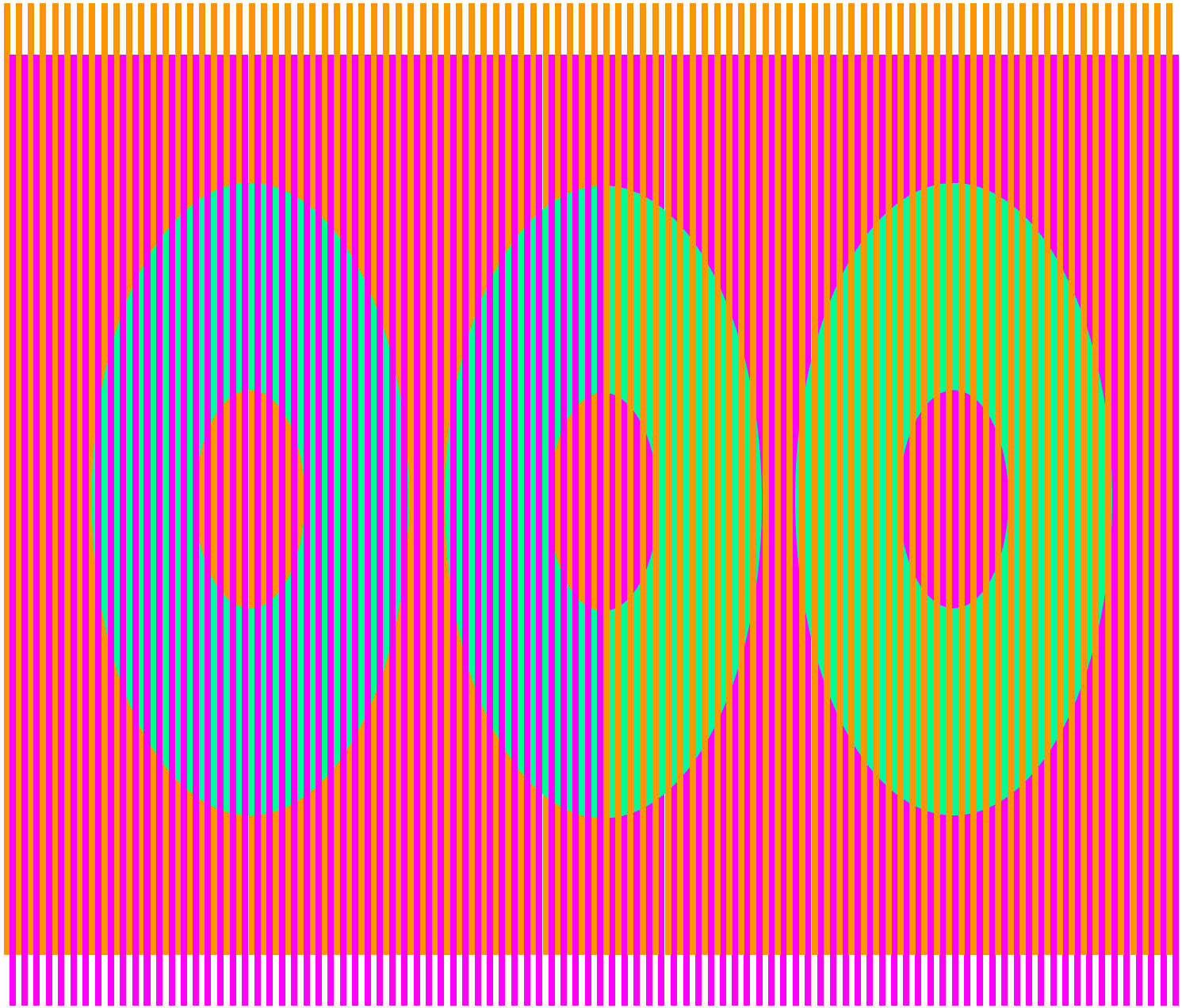




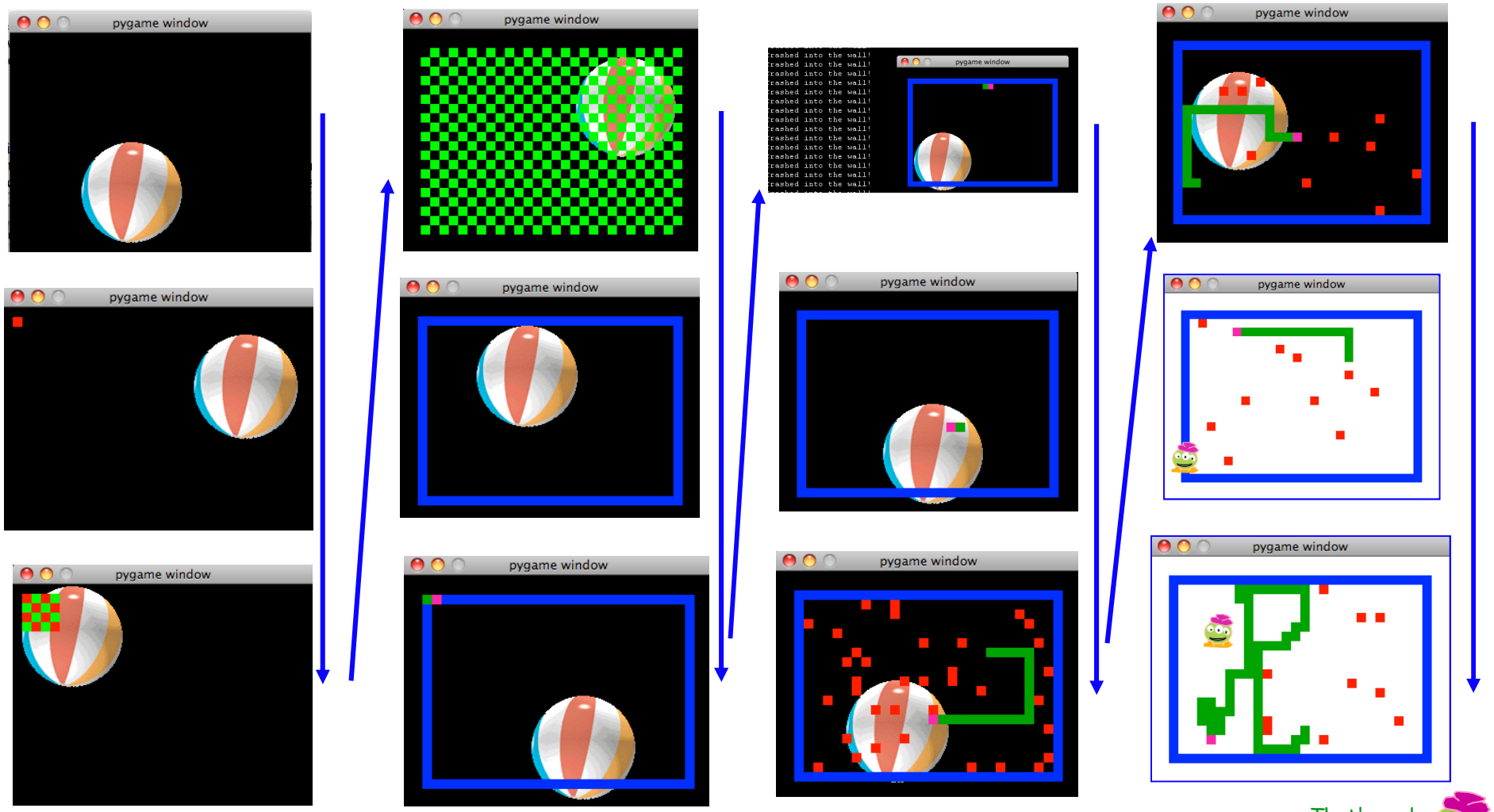








Other Courses?



Color definitions

matching based on pixel colors

Color-based matching...

red > 200

$|\text{hue}| < 0.1$

$|\text{hue}| < 0.1$

sat > 0.7

Future CS ?

Nostalgic for CS 5 itself ?

CS's Summer projects colloquium

Thursday, January 20, 2011 ~ place TBA

(but probably right here)

hw11pr2_smourad.py - C:\Users\dodds\Desktop\cs5_and_cgu_fall_2009\hw11pr2_smourad.py

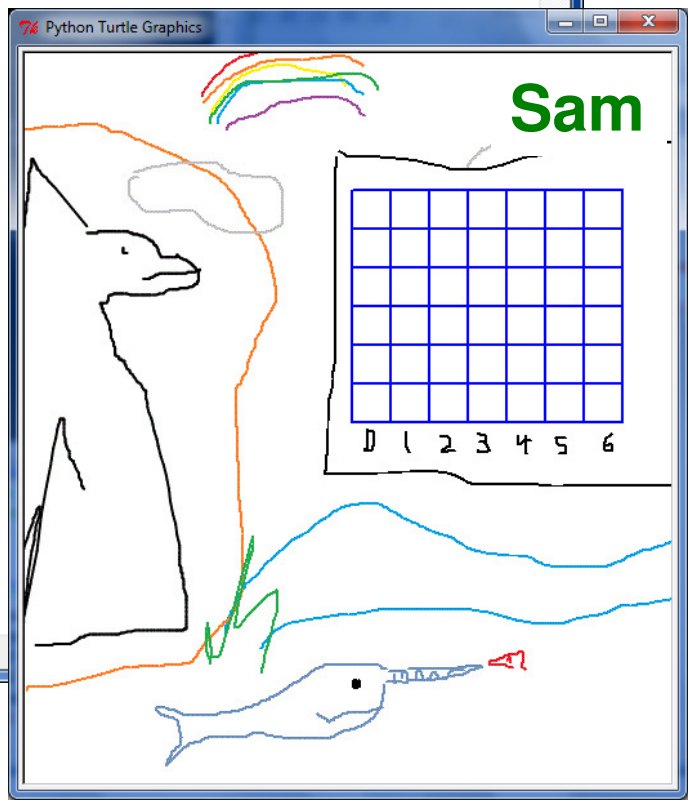
File Edit Format Run Options Windows Help

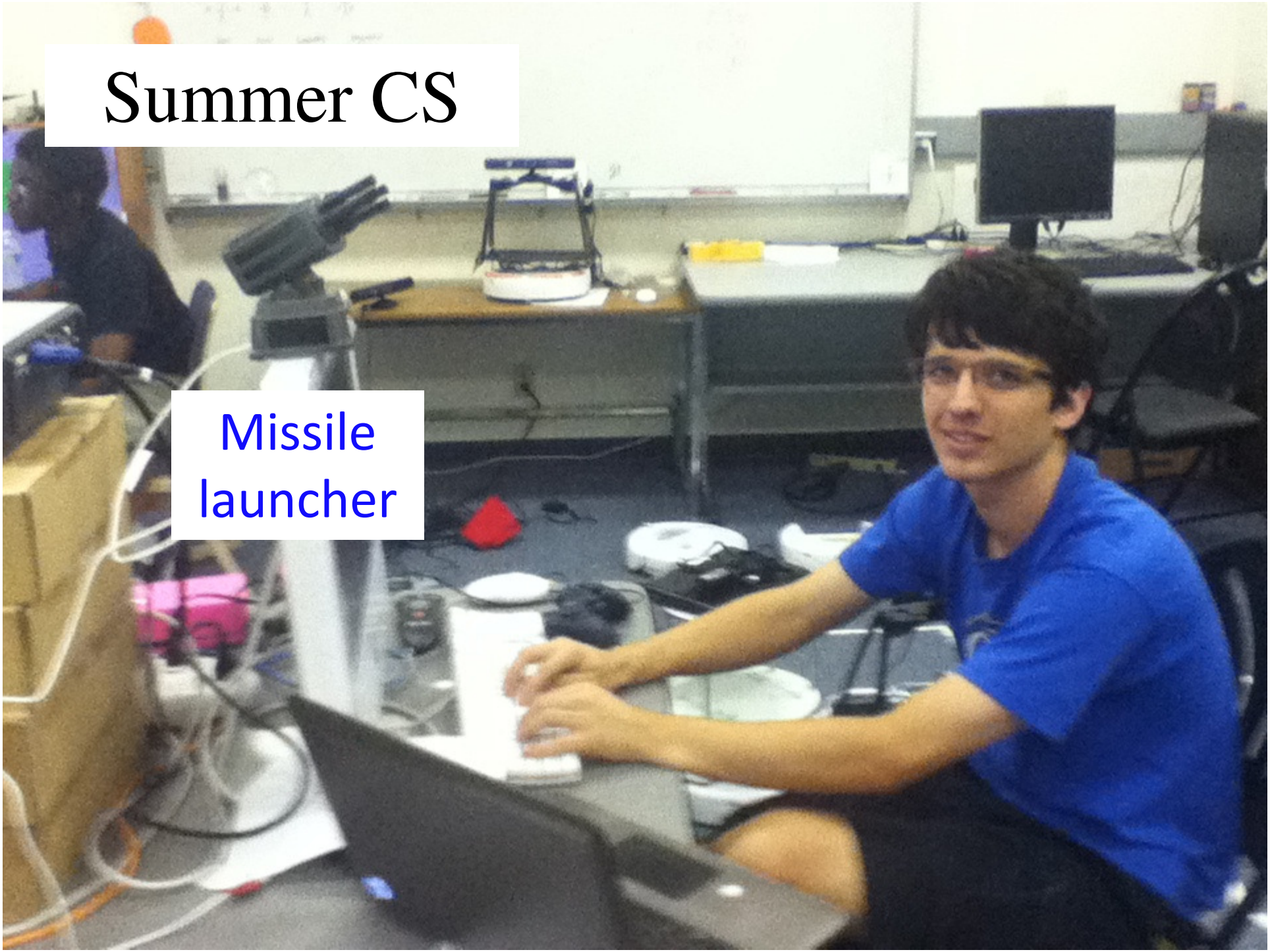
```
#
# Name: Sami Mourad. Height: 1m85cm. Weight: XXX. Color of eyes: brown.
# Color of hair: brown. Tel: 424-242-4242
#
# Dear HamsterBob, I am sorry to have deceived you with my foolishly
# negligent header comments. In an attempt to make up for my superciliousness,
# I have written a short story based on Goldie Locks and the three bears,
# and have incorporated it into my old connect four code to your (optional) enjoyment.

# Once upon a saturday morning...

# Big daddy bear wakes up from a dream filled with Cucurbita desserts.

from random import *

class Board:
    """ a datatype representing a C4 board
        with an arbitrary number of rows and cols
    """

    def __init__( self, width, height ):
        """ the constructor for objects of type Board """
        self.width = width
        # Bears love Cucurbita Maxima pies.
```

Grading CS5!

Sami

Python Turtle Graphics




Summer CS
Missile launcher


Summer CS
Robots '11
Phillies