UNIVERSITY OF
SOUTH CAROLINA

CSCE 574 ROBOTICS

Computer Vision

Slides curtesy of Professor Gregory Dudek

Ioannis Rekleitis

Why vision?

- Passive (emits nothing).
  - Discreet.
  - Energy efficient.
- Intuitive.
- Powerful (works well for us, right?)
- Long and short range.
- Fast.

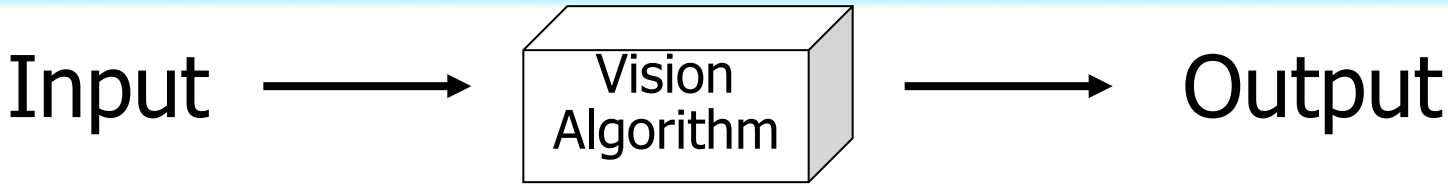
So, what's the problem?

- How hard is vision? Why do we think is do-able?

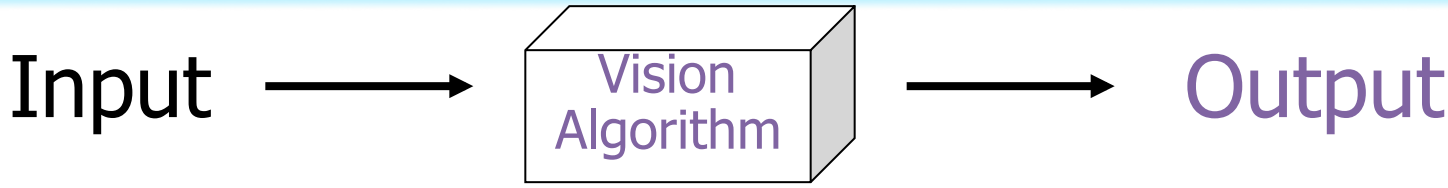
Problems:

- Slow.
- Data-heavy.
- Impossible.
- Mixes up many factors.

The "Vision Problem"



The "Vision Problem"



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

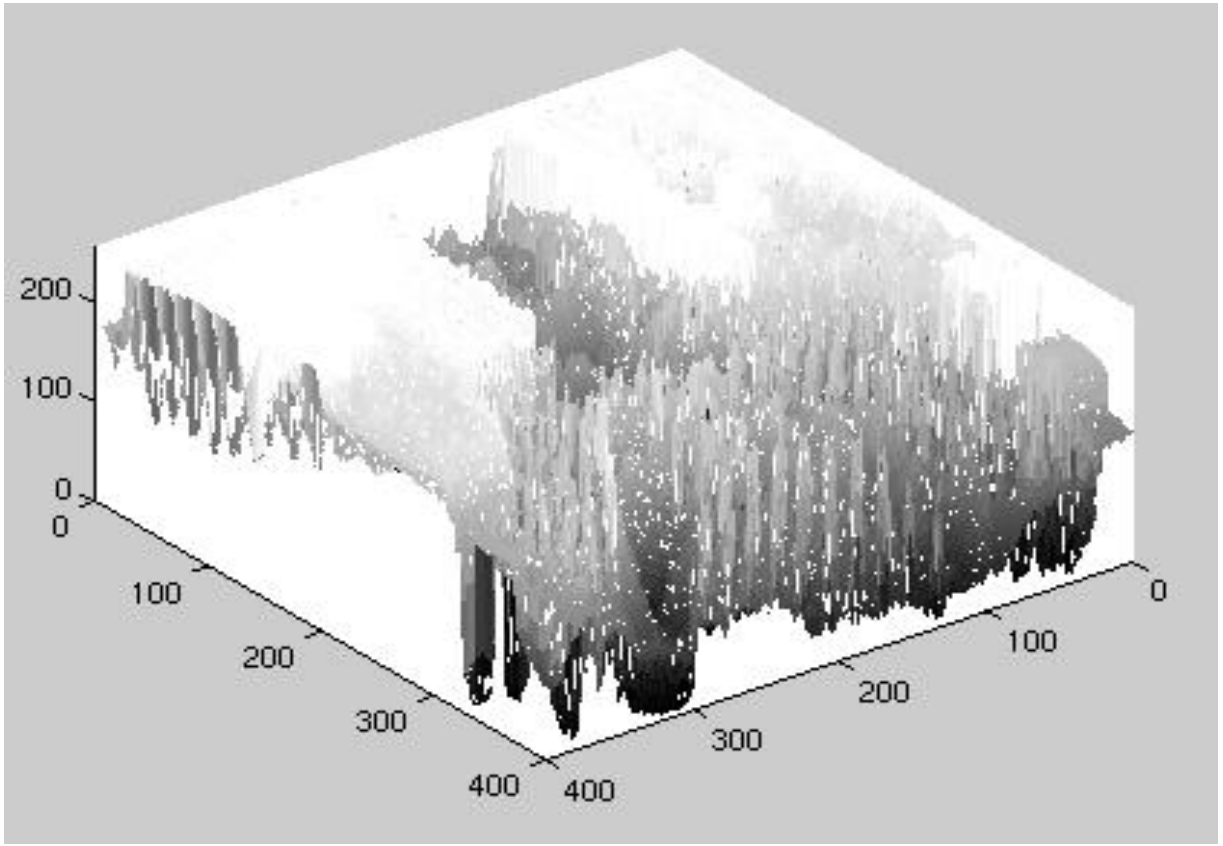
020 067 073 058 055 076 069 050 074 064 065 066 066 059 023
047 109 107 118 107 115 110 120 120 124 120 128 124 132 131
047 125 130 130 122 121 117 142 131 133 134 141 149 144 135
051 139 143 139 147 134 149 069 127 144 139 144 150 161 149
054 136 161 148 147 158 055 052 034 030 158 156 165 163 156
043 144 165 159 154 171 224 191 047 030 171 165 175 164 163
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

The "Vision Problem"

Input → Vision Algorithm → Output

207	200	194	194	203	130	105	095	107	153	192	196	190	186	175
242	194	222	254	255	124	074	082	072	076	208	206	202	194	185
254	170	204	255	248	122	153	135	111	081	252	253	233	232	250
255	172	201	255	249	123	092	040	094	106	255	253	239	150	254
255	197	192	255	248	133	024	027	076	032	250	255	255	181	255
253	210	190	239	250	089	092	149	128	013	254	254	253	229	255
252	238	180	218	251	106	116	181	140	024	250	255	255	255	200
248	248	169	227	252	111	066	118	061	021	252	251	255	255	142
253	255	171	254	253	142	037	132	006	017	253	253	255	254	201
253	250	170	255	253	139	134	127	156	078	255	253	253	254	237
254	228	169	213	235	146	123	096	090	130	230	250	253	254	254
252	244	140	215	245	125	055	043	081	077	252	234	253	253	253
254	250	169	211	235	117	108	093	119	078	246	249	235	225	255
254	234	167	212	217	110	070	049	098	074	244	246	239	207	254
255	219	170	238	253	113	130	109	063	075	243	235	233	252	252
255	221	179	248	227	111	083	041	061	083	240	249	243	232	253
221	217	180	213	243	109	079	048	100	045	246	249	244	221	210
236	216	178	208	230	156	077	062	110	088	244	249	230	220	221
229	224	183	211	132	052	087	062	124	085	135	246	236	220	214
230	223	185	185	112	079	008	124	158	125	119	119	232	225	232
221	215	194	100	154	071	008	031	097	010	093	098	148	229	216
223	217	132	046	072	076	056	048	013	182	073	076	083	215	219
224	216	041	102	090	162	079	111	118	164	083	170	065	221	219
215	222	046	111	077	075	060	046	069	032	179	068	157	224	226
219	216	092	045	074	143	013	171	159	072	087	065	143	217	222
222	224	070	041	074	131	085	150	112	140	139	154	055	231	218
226	232	118	109	041	165	130	105	097	175	078	081	067	064	174
253	254	079	072	116	089	020	068	103	074	031	130	106	052	161
047	034	090	045	145	027	135	109	082	082	048	113	087	061	157
193	192	057	038	051	092	018	062	110	052	060	084	066	071	154
191	192	043	153	052	030	078	061	062	054	046	049	054	078	158
184	181	066	019	043	038	046	083	057	050	145	048	035	087	158
138	074	030	082	030	038	076	041	141	046	045	040	009	063	149
135	016	057	071	035	025	040	062	030	084	130	043	059	113	151



The "Vision Problem"

Input → Vision Algorithm → Output

207 200 194 194 203 130 105 095 107 153 192 196 190 186 175
242 194 222 254 255 124 074 082 072 076 208 206 202 194 185
254 170 204 255 248 122 153 135 111 081 252 253 233 232 250
255 172 201 255 249 123 092 040 094 106 255 253 239 150 254
255 197 192 255 248 133 024 027 076 032 250 255 255 181 255
253 210 190 239 250 089 092 149 128 013 254 254 253 229 255
252 238 180 218 251 106 116 181 140 024 250 255 255 255 200
248 248 169 227 252 111 066 118 061 021 252 251 255 255 142
253 255 171 254 253 142 037 132 006 017 253 253 255 254 201
253 250 170 255 253 139 134 127 156 078 255 253 253 254 237
254 228 169 213 235 146 123 096 090 130 230 250 253 254 254
252 244 140 215 245 125 055 043 081 077 252 234 253 253 253
254 250 169 211 235 117 108 093 119 078 246 249 235 225 255
254 234 167 212 217 110 070 049 098 074 244 246 239 207 254
255 219 170 238 253 113 130 109 063 075 243 235 233 252 252
255 221 179 248 227 111 083 041 061 083 240 249 243 232 253
221 217 180 213 243 109 079 048 100 045 246 249 244 221 210
236 216 178 208 230 156 077 062 110 088 244 249 230 220 221
229 224 183 211 132 052 087 062 124 085 135 246 236 220 214
230 223 185 185 112 079 008 124 158 125 119 119 232 225 232
221 215 194 100 154 071 008 031 097 010 093 098 148 229 216
223 217 132 046 072 076 056 048 013 182 073 076 083 215 219
224 216 041 102 090 162 079 111 118 164 083 170 065 221 219
215 222 046 111 077 075 060 046 069 032 179 068 157 224 226
219 216 092 045 074 143 013 171 159 072 087 065 143 217 222
222 224 070 041 074 131 085 150 112 140 139 154 055 231 218
226 232 118 109 041 165 130 105 097 175 078 081 067 064 174
253 254 079 072 116 089 020 068 103 074 031 130 106 052 161
047 034 090 045 145 027 135 109 082 082 048 113 087 061 157
193 192 057 038 051 092 018 062 110 052 060 084 066 071 154
191 192 043 153 052 030 078 061 062 054 046 049 054 078 158
184 181 066 019 043 038 046 083 057 050 145 048 035 087 158
138 074 030 082 030 038 076 041 141 046 045 040 009 063 149
135 016 057 071 035 025 040 062 030 084 130 043 059 113 151

The "Vision Problem"

Input → Vision Algorithm → Output

207	200	194	194	203	130	105	095	107	153	192	196	190	186	175
242	194	222	254	255	124	074	082	072	076	208	206	202	194	185
254	170	204	255	248	122	153	135	111	081	252	253	233	232	250
255	172	201	255	249	123	092	040	094	106	255	253	239	150	254
255	197	192	255	248	133	024	027	076	032	250	255	255	181	255
253	210	190	239	250	089	092	149	128	013	254	254	253	229	255
252	238	180	218	251	106	116	181	140	024	250	255	255	255	200
248	248	169	227	252	111	066	118	061	021	252	251	255	255	142
253	255	171	254	253	142	037	132	006	017	253	253	255	254	201
253	250	170	255	253	139	134	127	156	078	255	253	253	254	237
254	228	169	213	235	146	123	096	090	130	230	250	253	254	254
252	244	140	215	245	125	055	043	081	077	252	234	253	253	253
254	250	169	211	235	117	108	093	119	078	246	249	235	225	255
254	234	167	212	217	110	070	049	098	074	244	246	239	207	254
255	219	170	238	253	113	130	109	063	075	243	235	233	252	252
255	221	179	248	227	111	083	041	061	083	240	249	243	232	253
221	217	180	213	243	109	079	048	100	045	246	249	244	221	210
236	216	178	208	230	156	077	062	110	088	244	249	230	220	221
229	224	183	211	132	052	087	062	124	085	135	246	236	220	214
230	223	185	185	112	079	008	124	158	125	119	119	232	225	232
221	215	194	100	154	071	008	031	097	010	093	098	148	229	216
223	217	132	046	072	076	056	048	013	182	073	076	083	215	219
224	216	041	102	090	162	079	111	118	164	083	170	065	221	219
215	222	046	111	077	075	060	046	069	032	179	068	157	224	226
219	216	092	045	074	143	013	171	159	072	087	065	143	217	222
222	224	070	041	074	131	085	150	112	140	139	154	055	231	218
226	232	118	109	041	165	130	105	097	175	078	081	067	064	174
253	254	079	072	116	089	020	068	103	074	031	130	106	052	161
047	034	090	045	145	027	135	109	082	082	048	113	087	061	157
193	192	057	038	051	092	018	062	110	052	060	084	066	071	154
191	192	043	153	052	030	078	061	062	054	046	049	054	078	158
184	181	066	019	043	038	046	083	057	050	145	048	035	087	158
138	074	030	082	030	038	076	041	141	046	045	040	009	063	149
135	016	057	071	035	025	040	062	030	084	130	043	059	113	151

The "Vision Problem"

Input → Vision 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

What does a robot need ?

doesn't need a full interpretation of available images

"This is Prof. X in his office offering me a cup of iced tea."

does need information about what to do...

"Run Away!!"

reactive

deliberative

- avoiding obstacles (or predators)
- pursuing objects
- localizing itself
- Mapping
- finding targets
- reasoning about the world ...

environmental interactions

What does a robot need ?

- What a camera does to the 3d world...

Shigeo Fukuda

squeezes away one dimension

http://www.psychologie.tu-dresden.de/i1/kaw/diverses Material/www.illusionworks.com/html/art_of_shigeo_fukuda.html

What does a robot need ?

- What a camera does to the 3d world...

Shigeo Fukuda

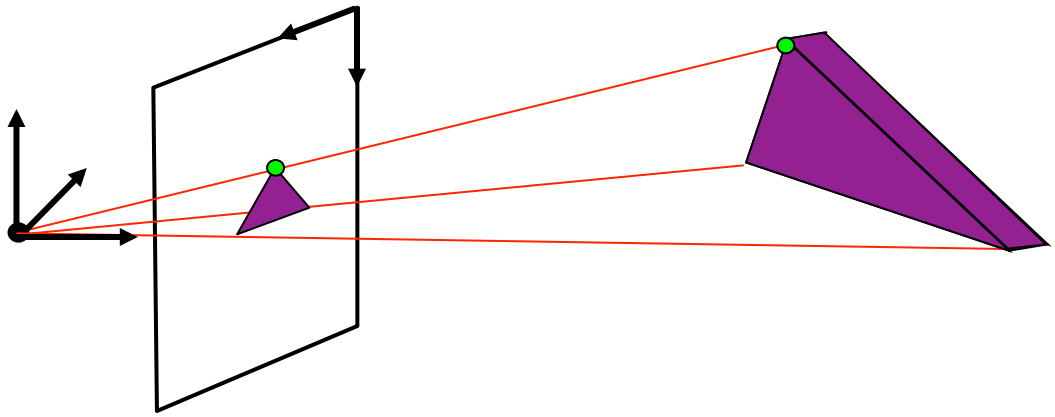
http://www.psychologie.tu-dresden.de/i1/kaw/diverses Material/www.illusionworks.com/html/art_of_shigeo_fukuda.html

Ill-posed

- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.

The vision problem in general…

- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.
- Basically, there are too many possible worlds that might (in theory) give rise to a particular image



Ill-posed

- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.

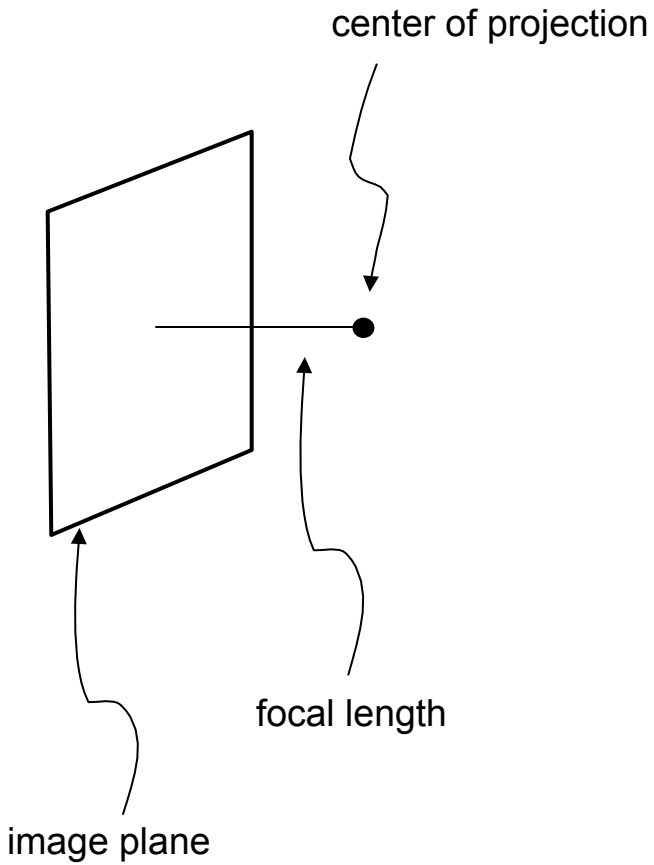
Ill-posed

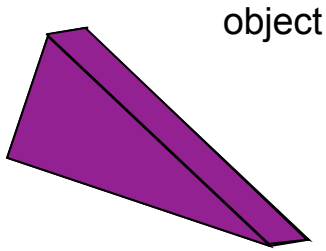
- In trying to extract 3d structure from 2d images, vision is an *ill-posed* problem.

  - An image isn't enough to disambiguate the many possible 3d worlds that could have produced it.

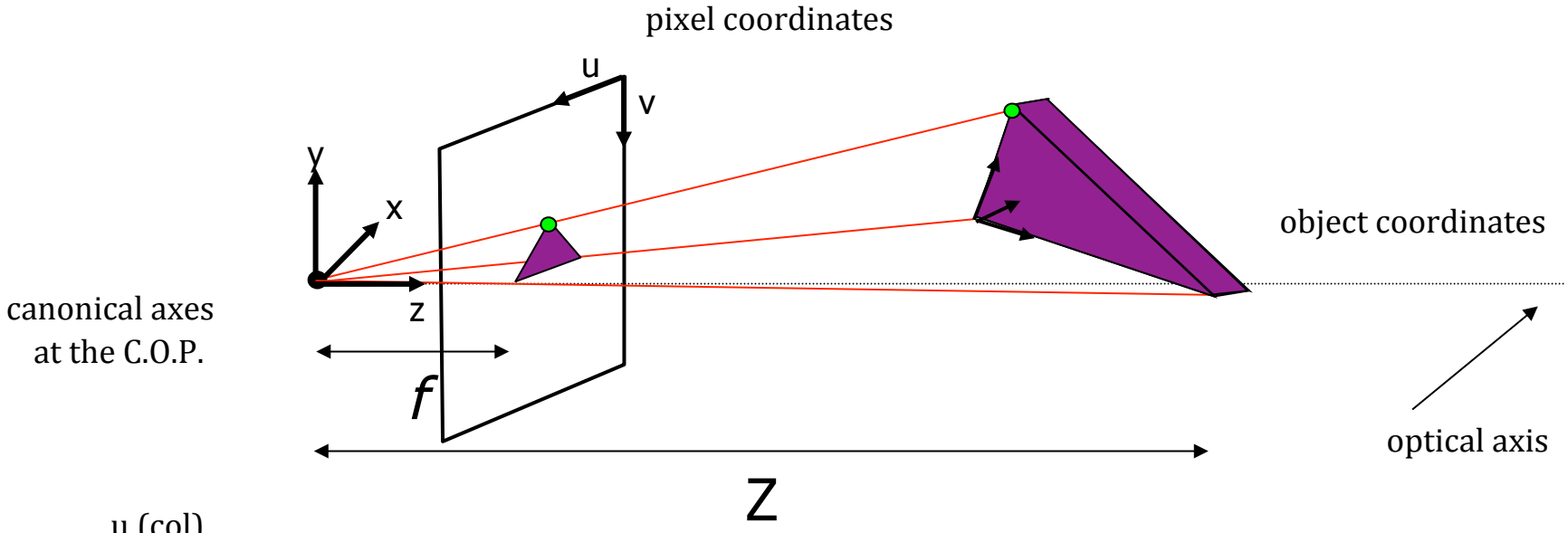
Camera Geometry

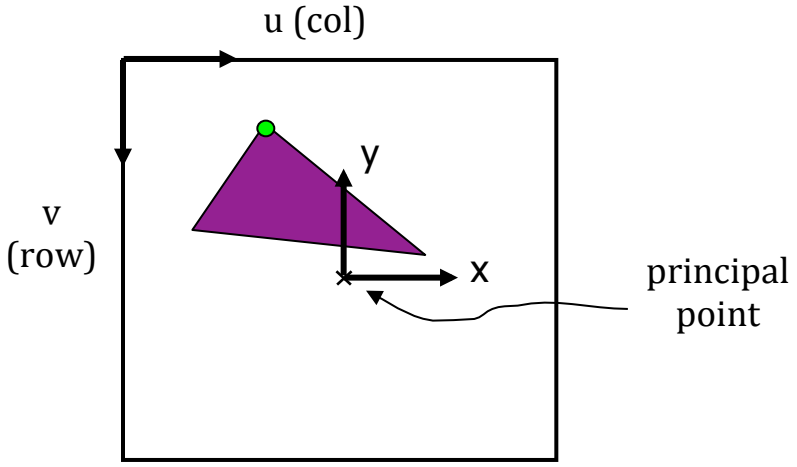
3D→2D transformation: perspective projection





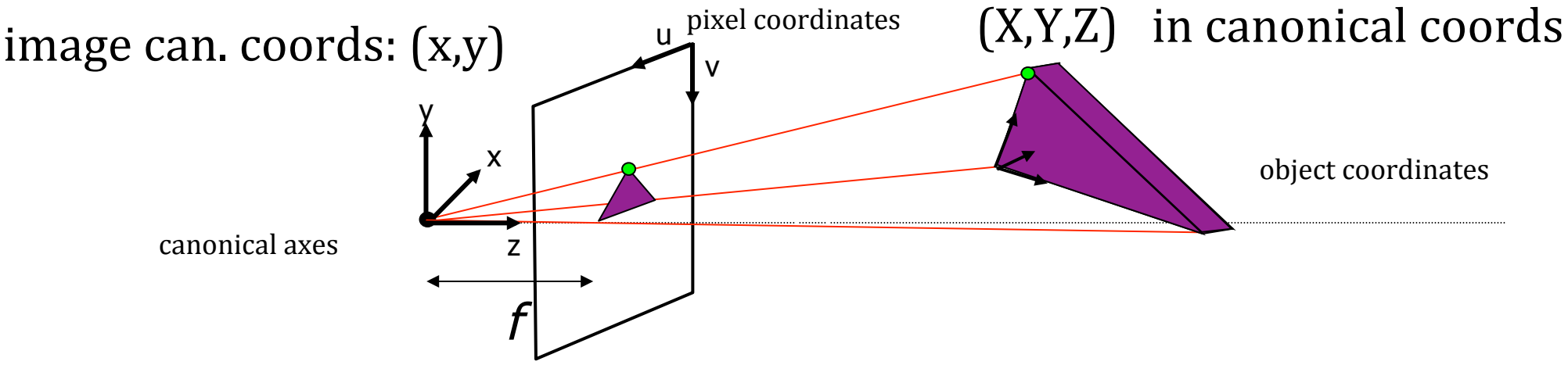
Coordinate Systems



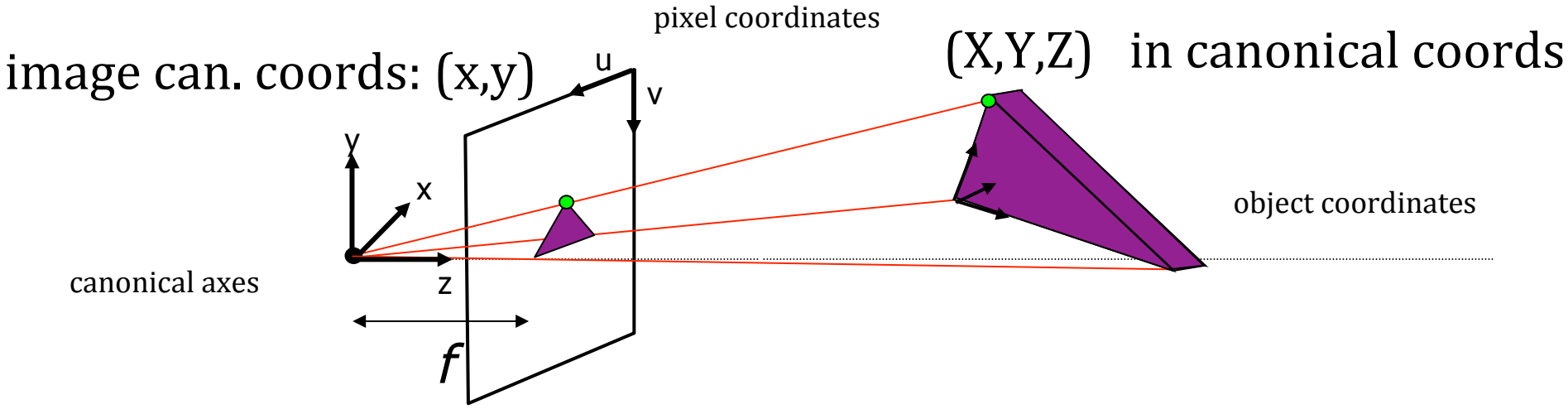


Add coordinate systems in order to describe feature points...

Coordinate Systems



From 3d to 2d



$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z}$$

a nonlinear transformation

goal: to recover information about (X,Y,Z) from (x,y)

Camera Calibration

- Camera Model
  - $[u\ v\ 1]$ Pixel coords
  - $\begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix}^T$ World coords

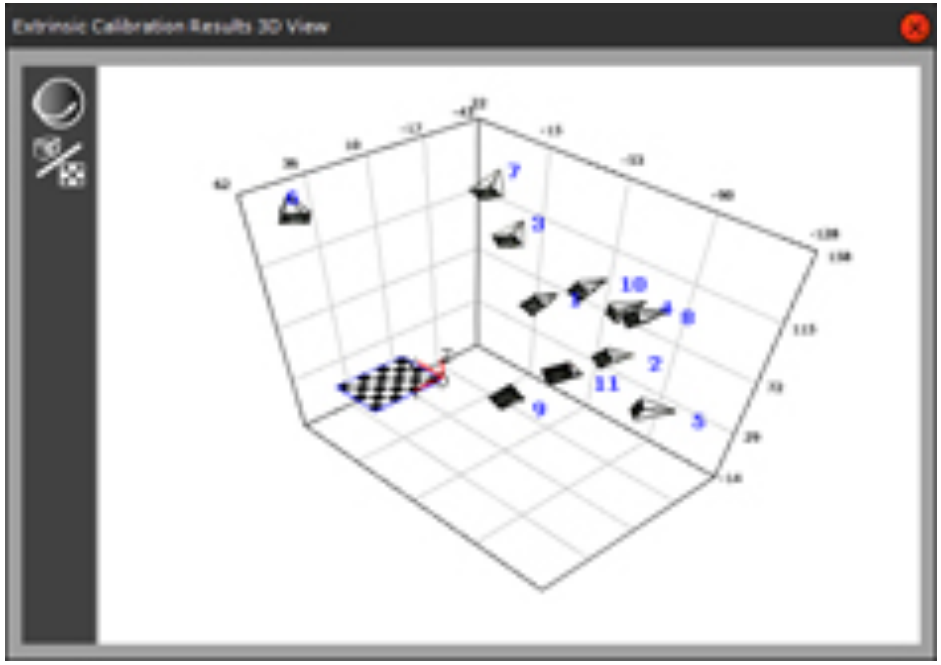
$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

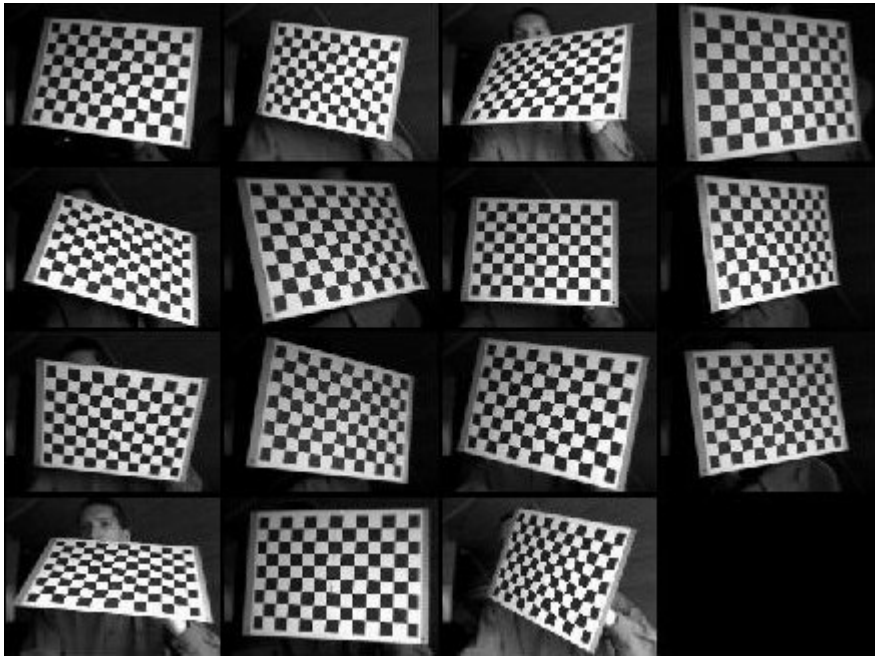
- Intrinsic Parameters
  - $\alpha_x = f \cdot m_x, \alpha_y = f \cdot m_y$ focal lengths in pixels
  - γ skew coefficient
  - u_0, v_o focal point

$$A = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

- Extrinsic Parameters
  - $\begin{bmatrix} R & T \end{bmatrix}$ Rotation and Translation

Camera Calibration





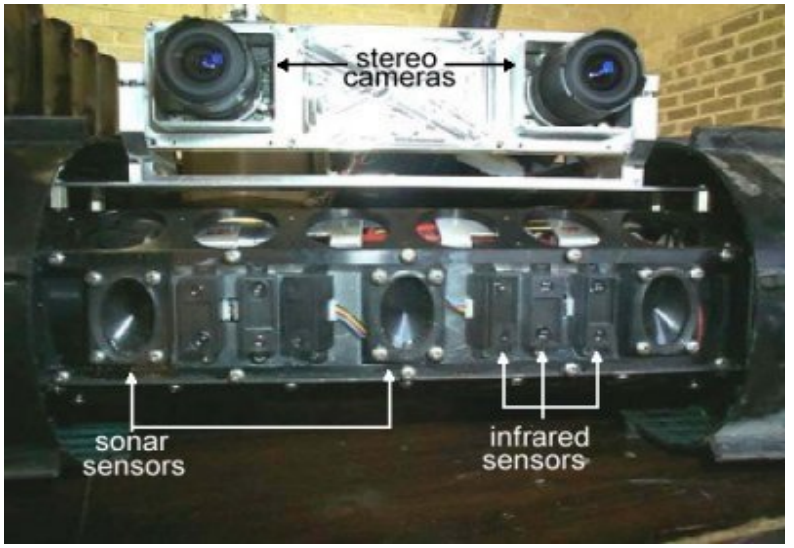
Existing packages in MATLAB, OpenCV, etc

A Vision "solution"

- If interpreting a single image is difficult... What about more ?!

multiple times

multiple cameras



Robot vision sampler

A brief overview of robotic vision processing...

(1) Image streams → simplified via generality
(1) Image streams → simplified via specificity

(2) Stereo vision → (or beyond...)

(3) Incorporating vision within robot control
- ↓ 3d reconstruction
- ↓ Visual "servoing"

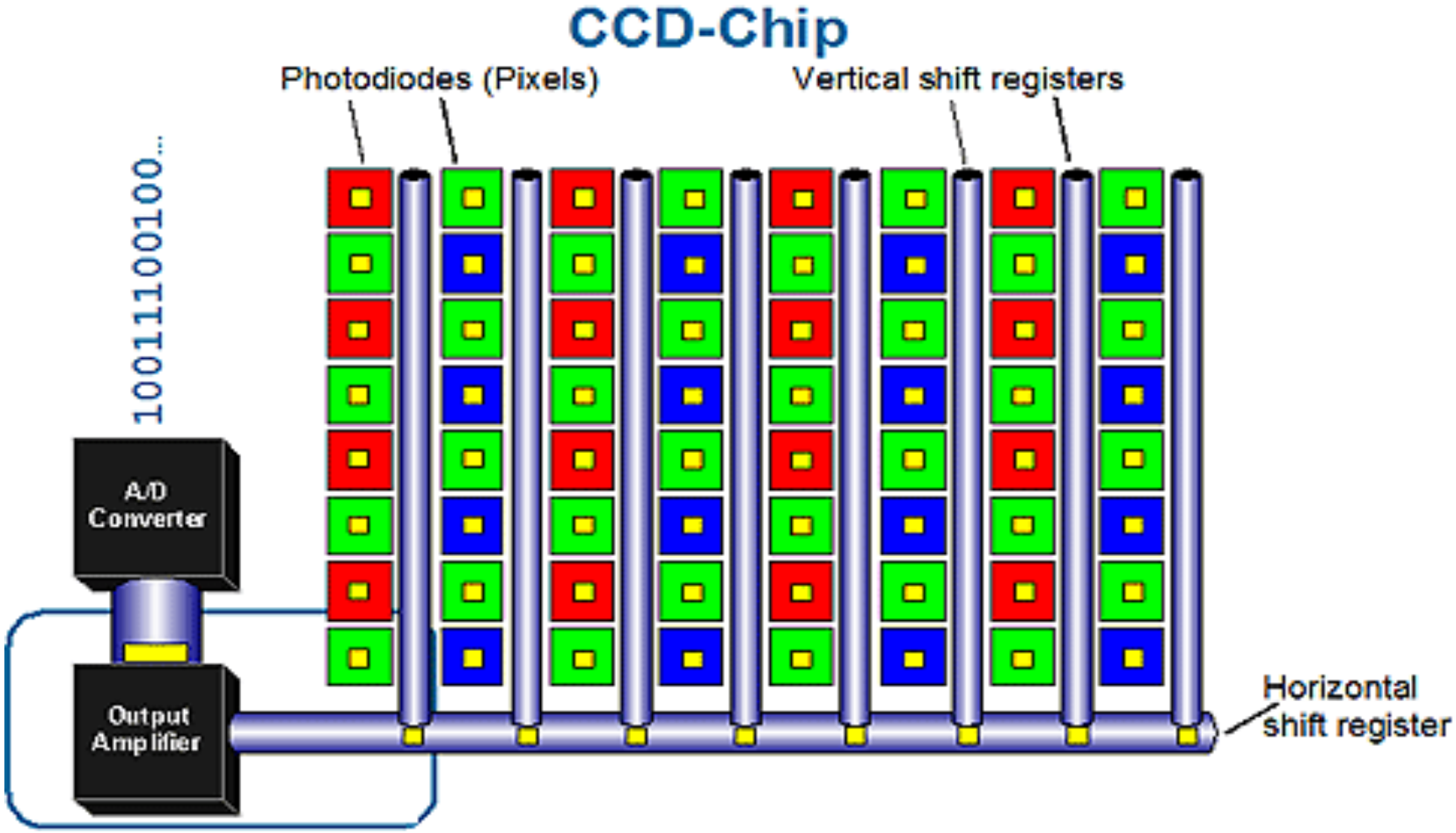
speaking of servoing...

Visual Servoing

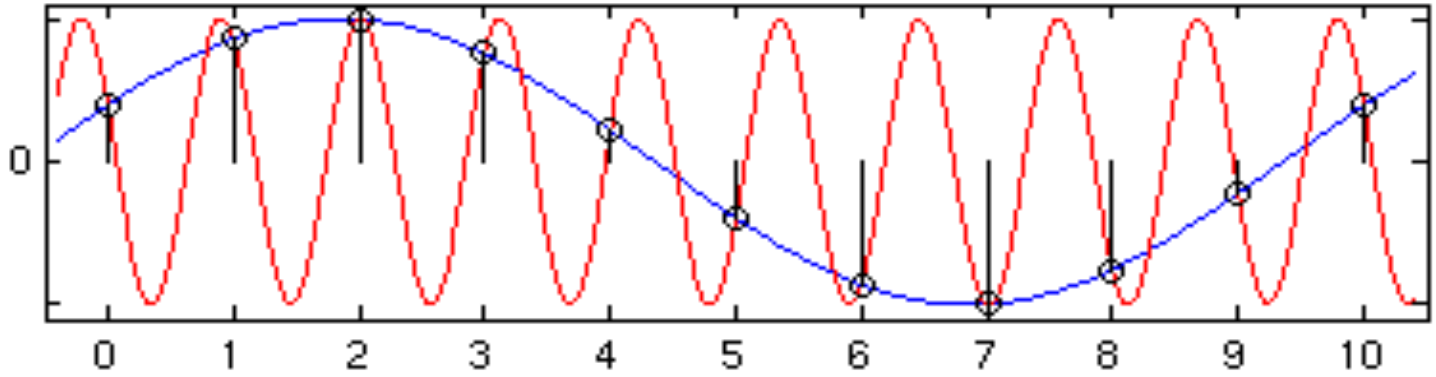
Details

- Images are not actually continuous.
- The sampling (and hardware) issues lead to a few other minor problems.

CCD (Charge-Coupled Device)



Aliasing.



- To avoid: $f_{sampling} > 2F_{max}$
  - Nyquist Rate

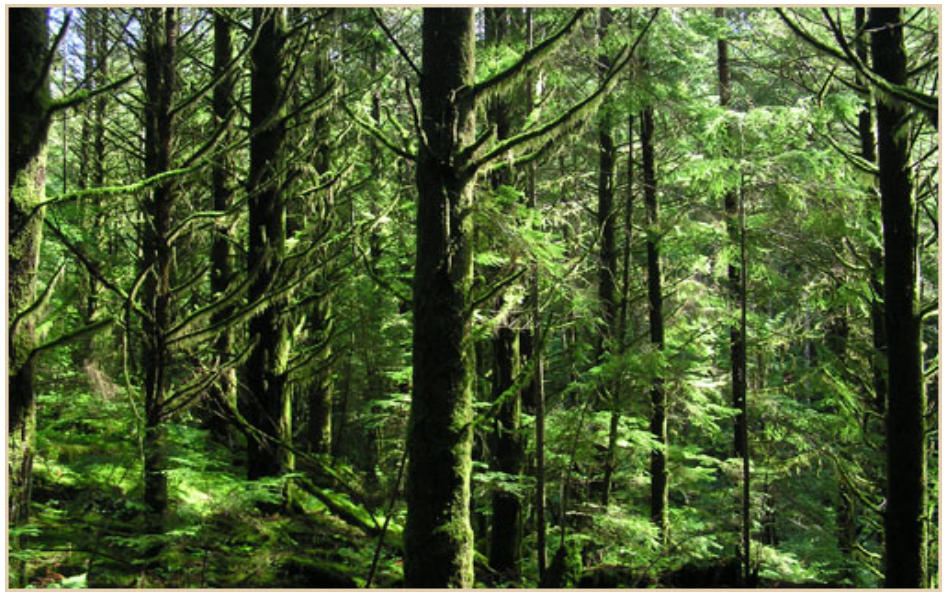
Aliasing: Moiré Patterns

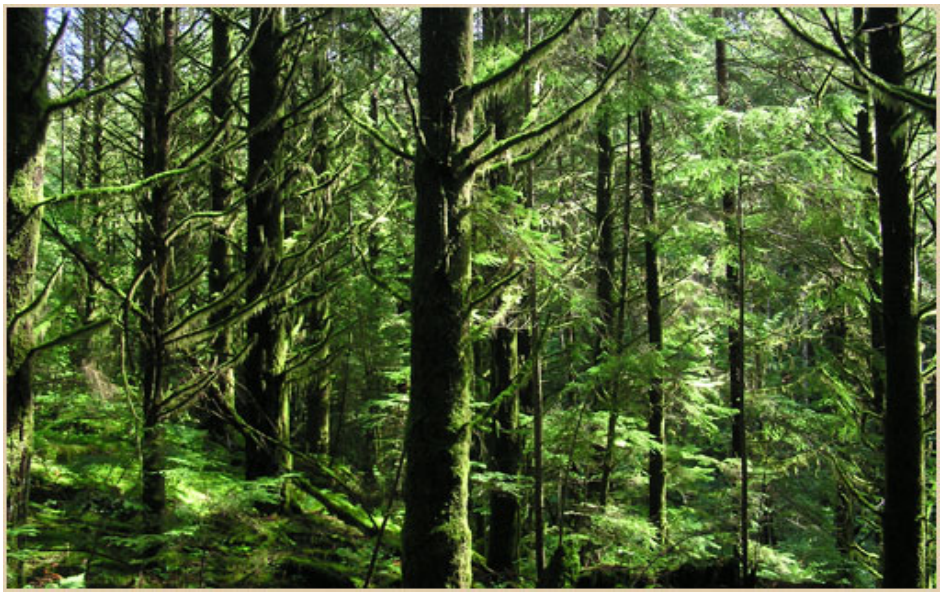
Key problems

- Recognition:
  - What is that thing in the picture?
  - What are all the things in the image?
- Scene interpretation
  - Describe the image?
- Scene "reconstruction":
  - What is the 3-dimensional layout of the scene?
  - What are the physical parameters that gave rise to the image?
  - What is a description of the scene?

Notion of an "inverse problem."

Correspondence Problem



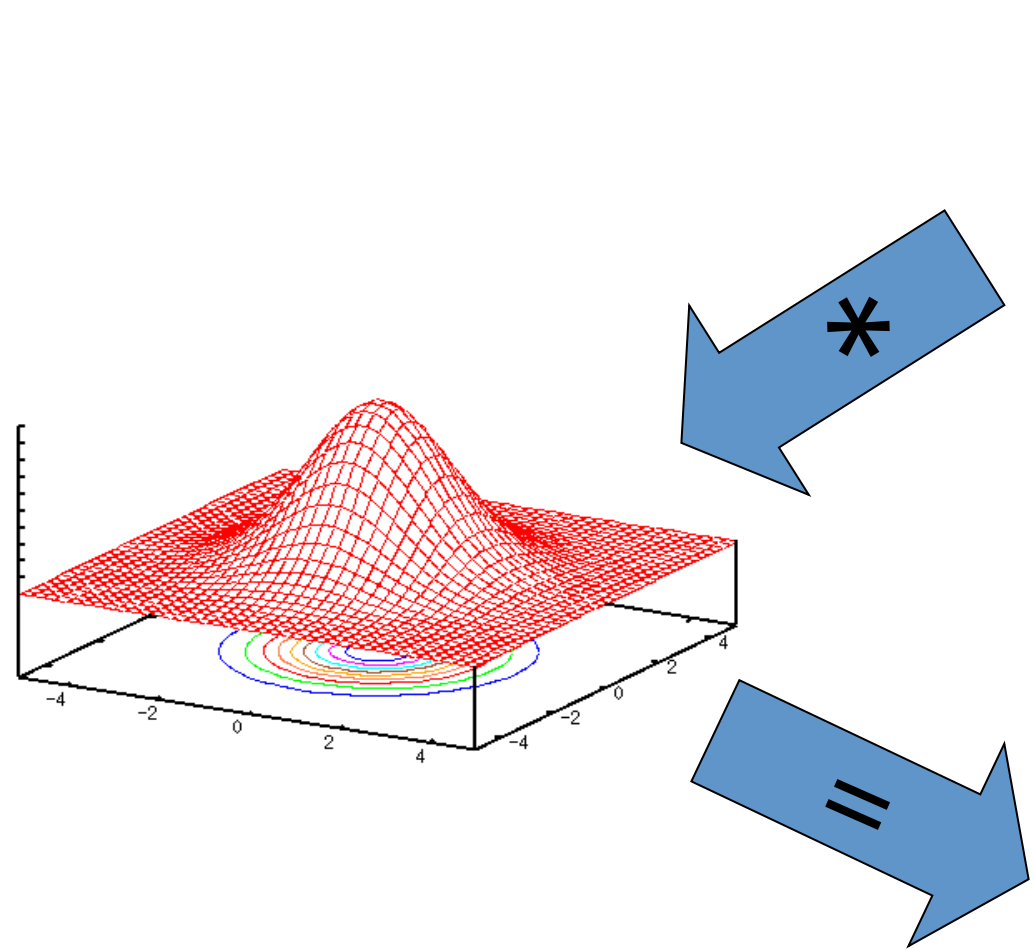


Correspondence

From I_1

From I_2

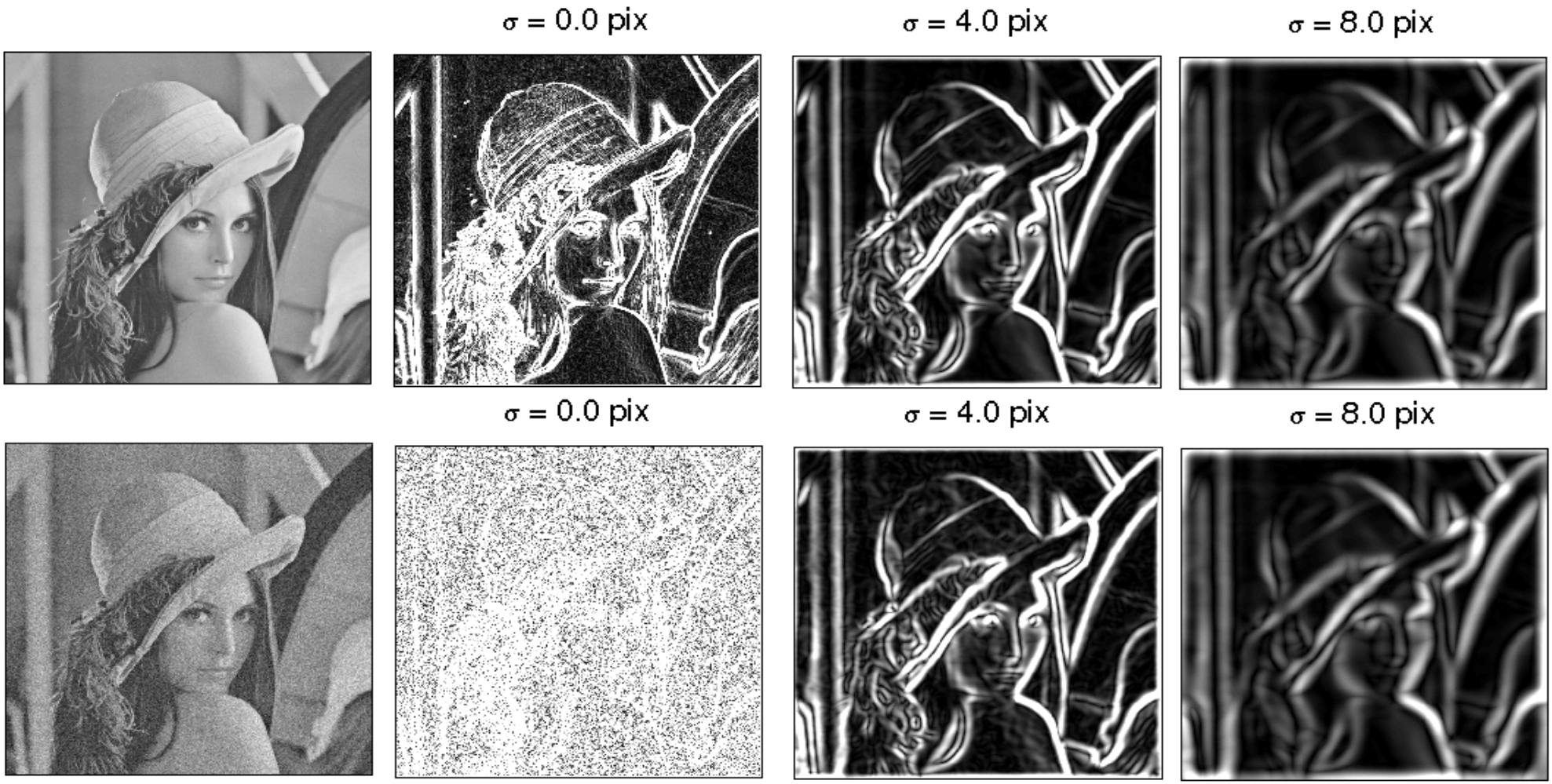
Gaussian Blur



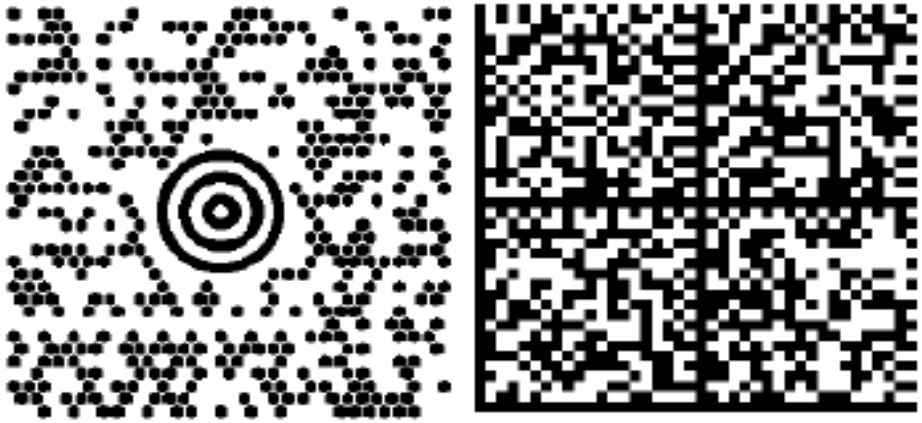
Gaussian Blur and Noise

Gaussian Blur and Noise

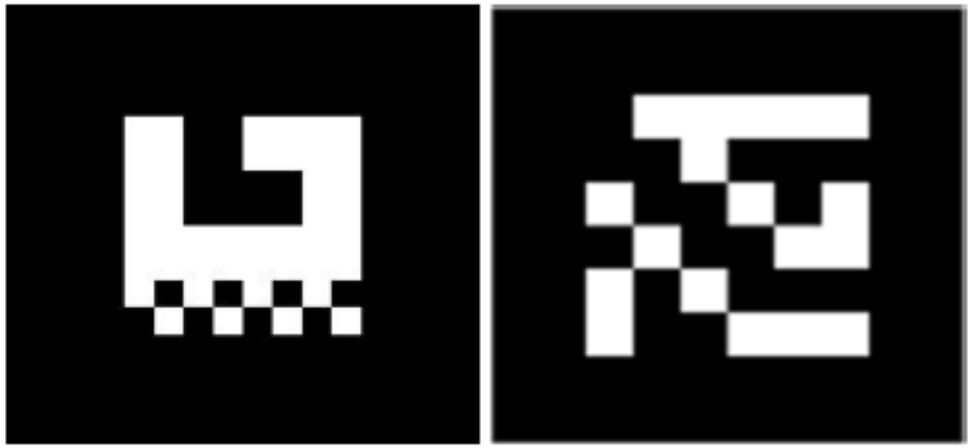
Gaussian Blur, Noise, Sobel



Fiduciary Markers/Fiducial



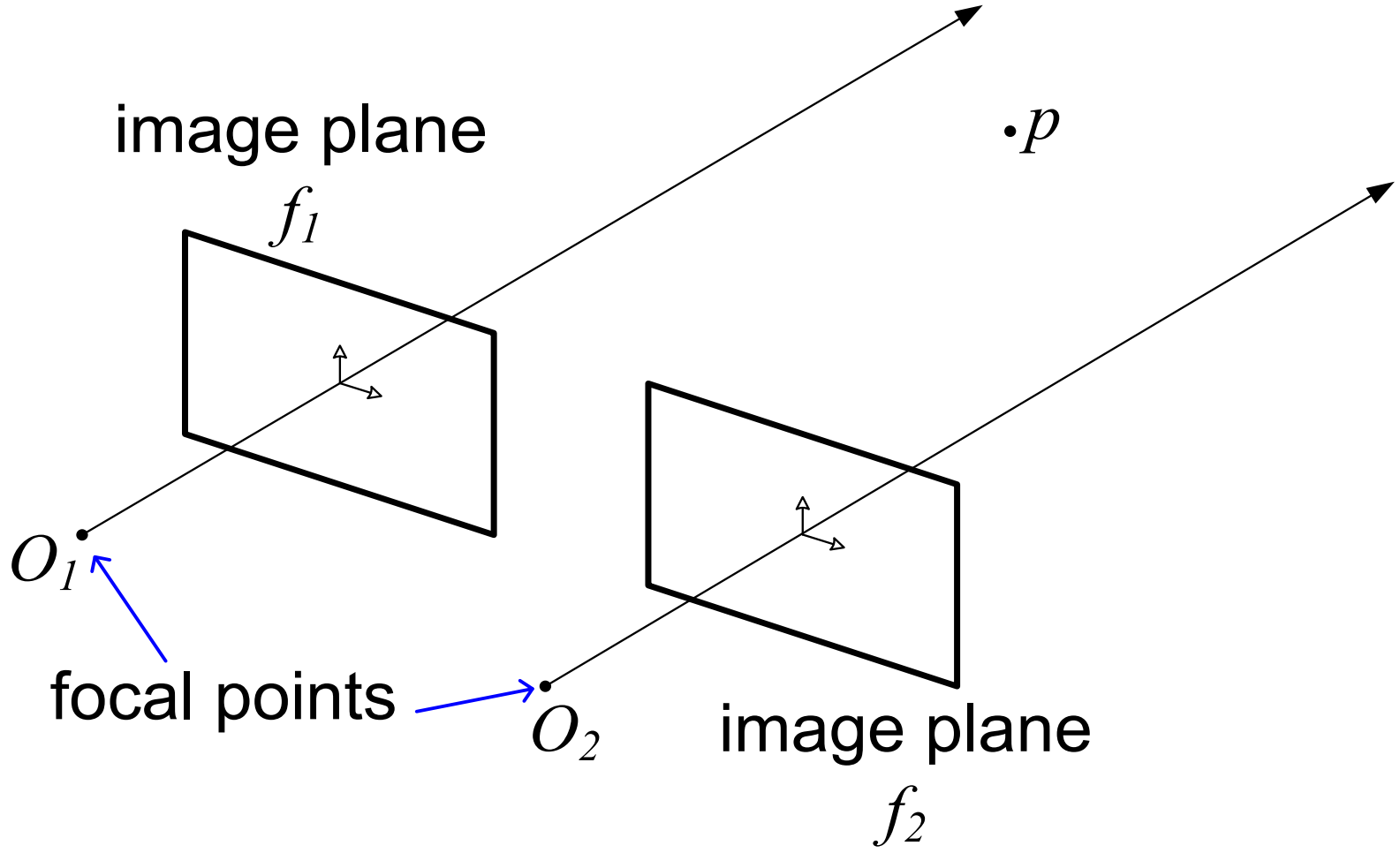
(a) MaxiCode (b) DataMatrixSymbol



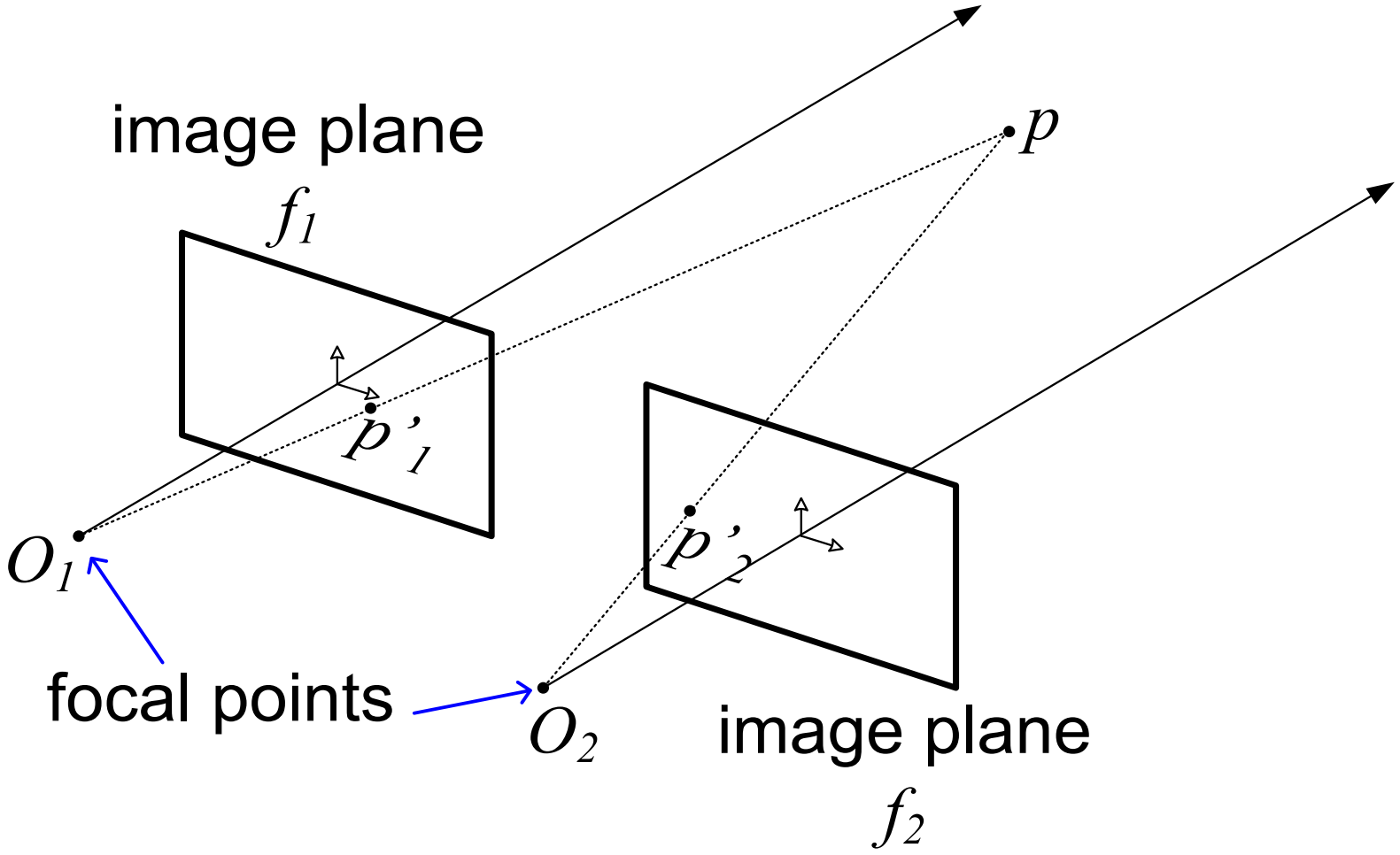
(c) ARToolkit (d) ARTag

Fourier Tag

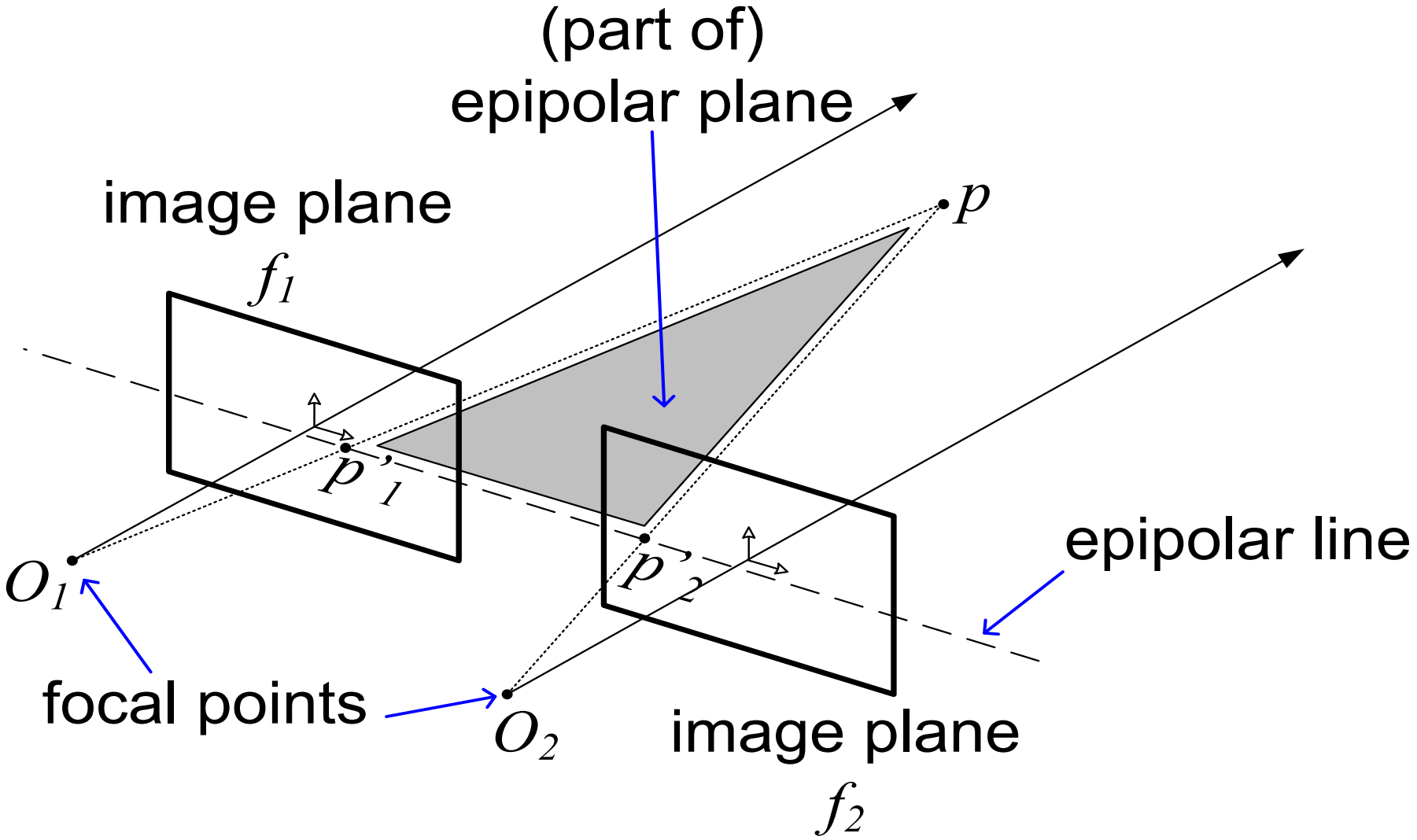
Stereo Vision: Pinhole Camera



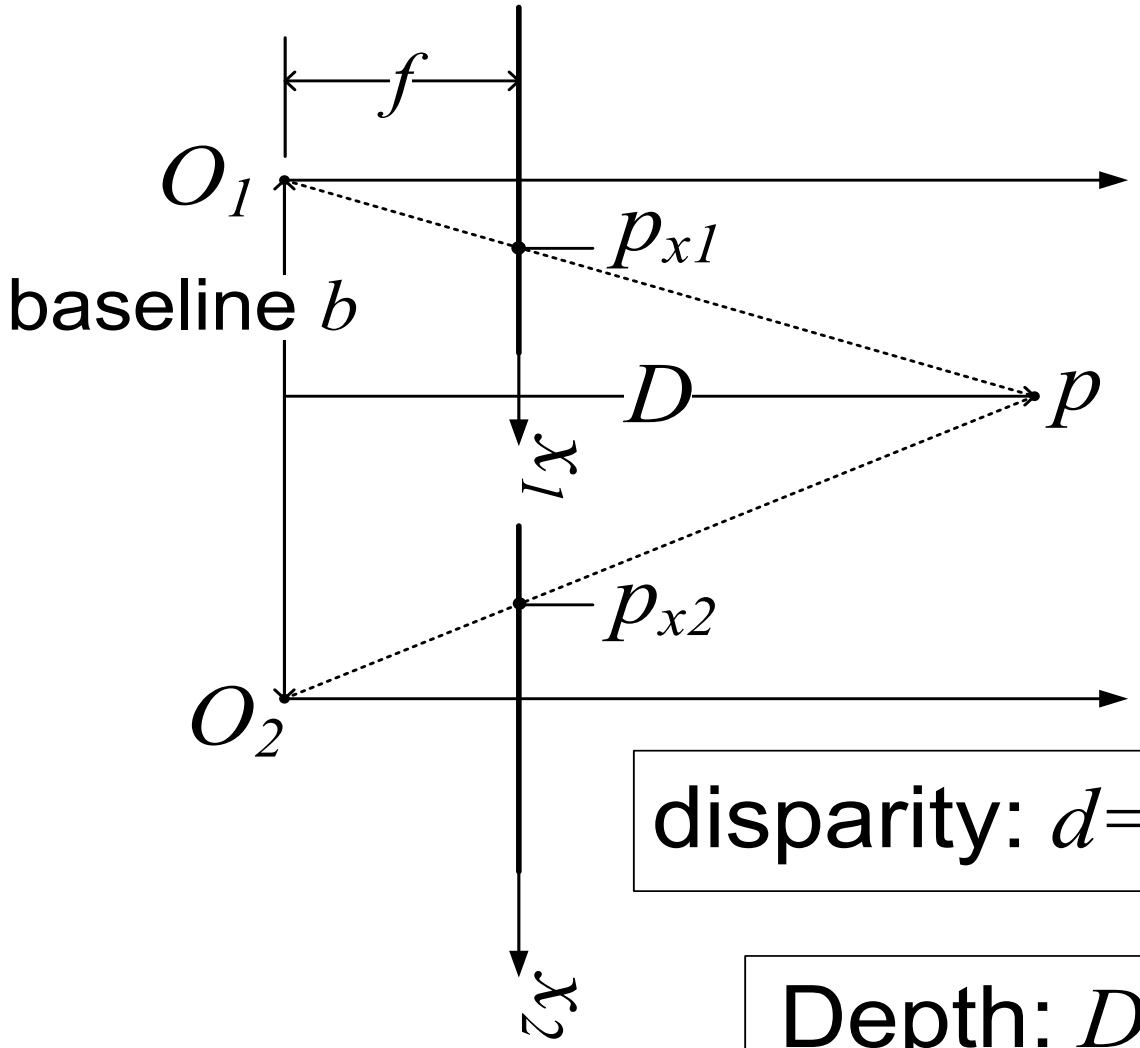
Stereo Vision: Pinhole Camera



Stereo Vision: Pinhole Camera



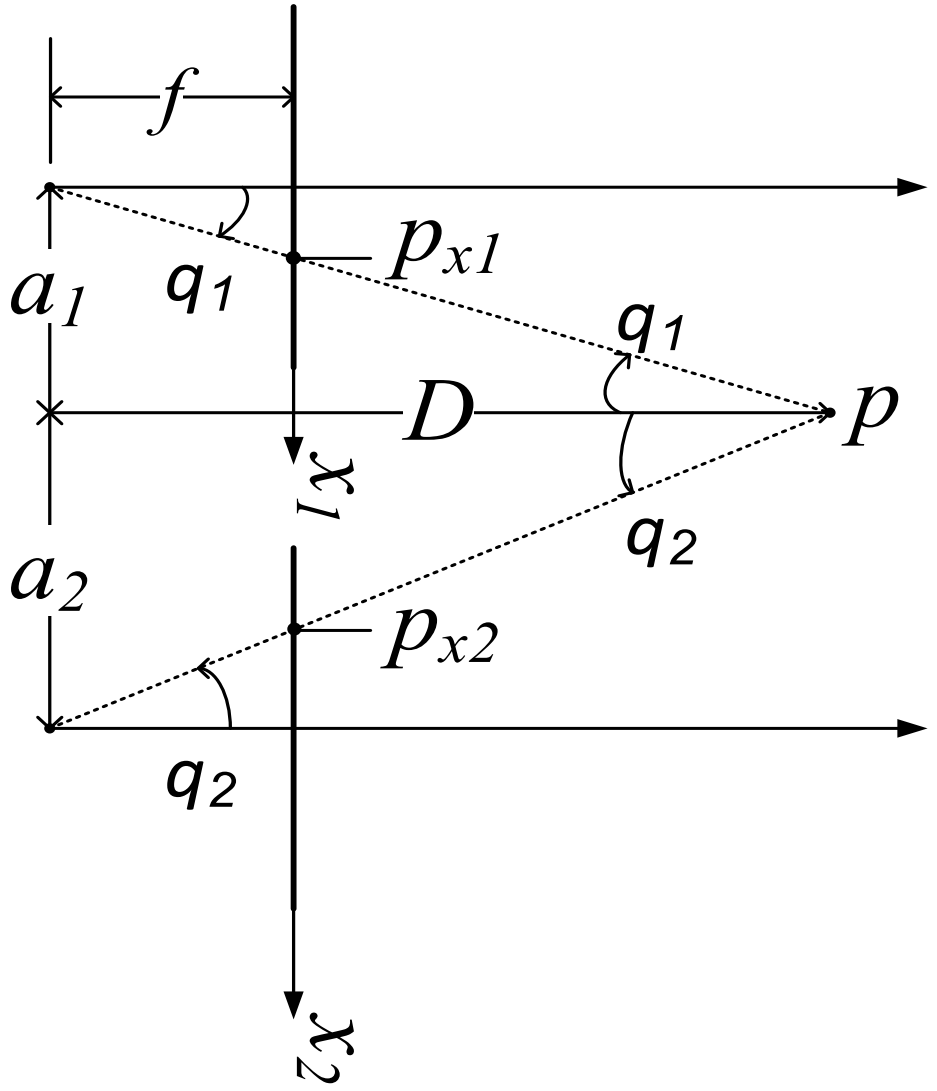
Stereo Vision: Pinhole



disparity: $d=p_{x1}-p_{x2}$

Depth: $D=fb/d$

Stereo Vision: Pinhole



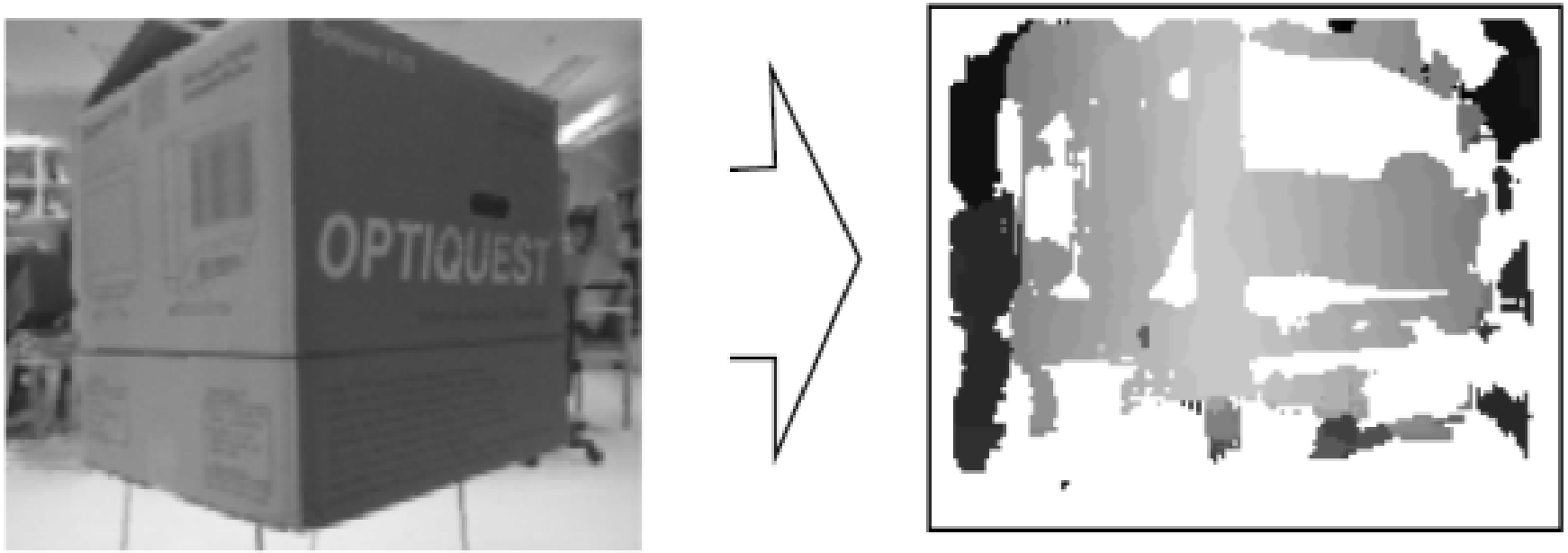
$$\frac{p_{x1}}{f} = \frac{a_1}{D}$$

$$\frac{p_{x2}}{f} = \frac{a_2}{D}$$

$$a_1 + a_2 = b$$

Large Baseline

Stereo: Disparity Map

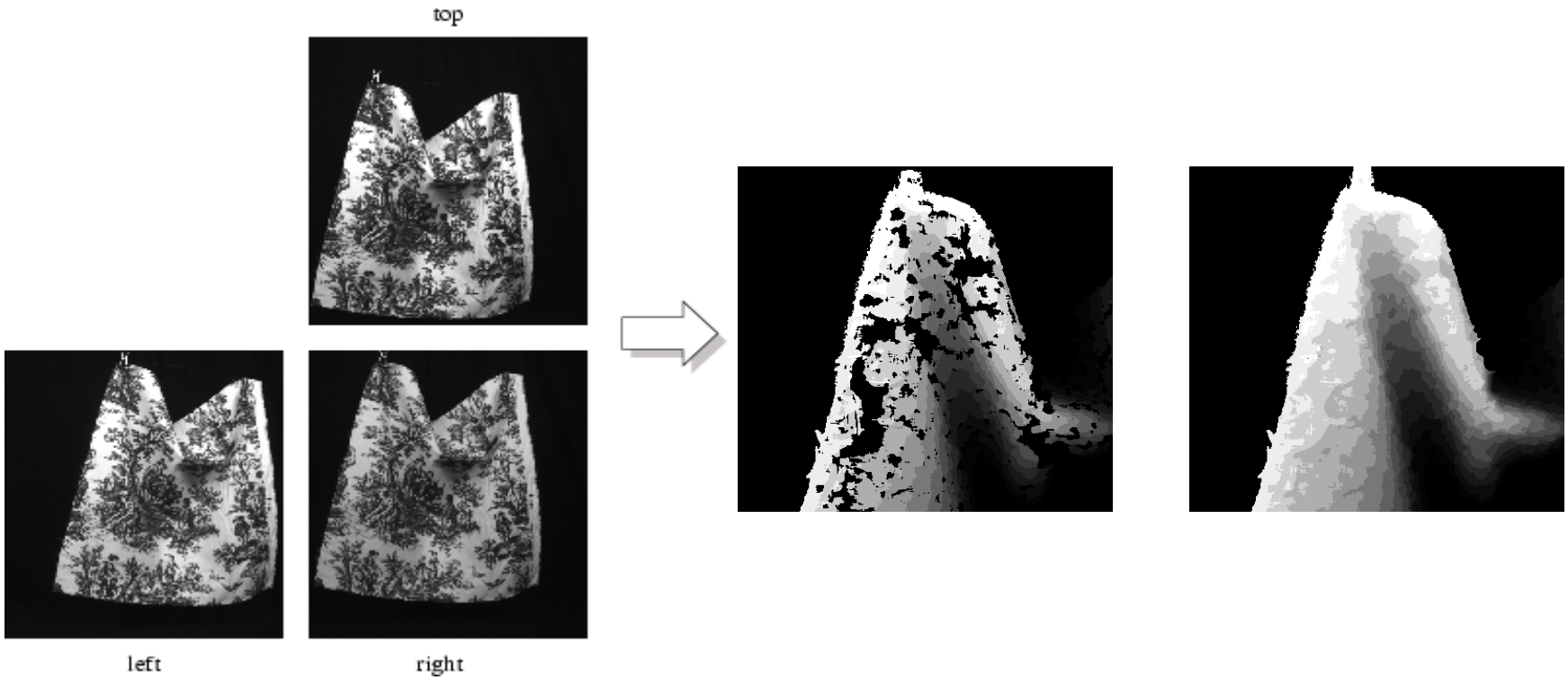


Using real-time stereo vision for mobile robot navigation

Don Murray Jim Little

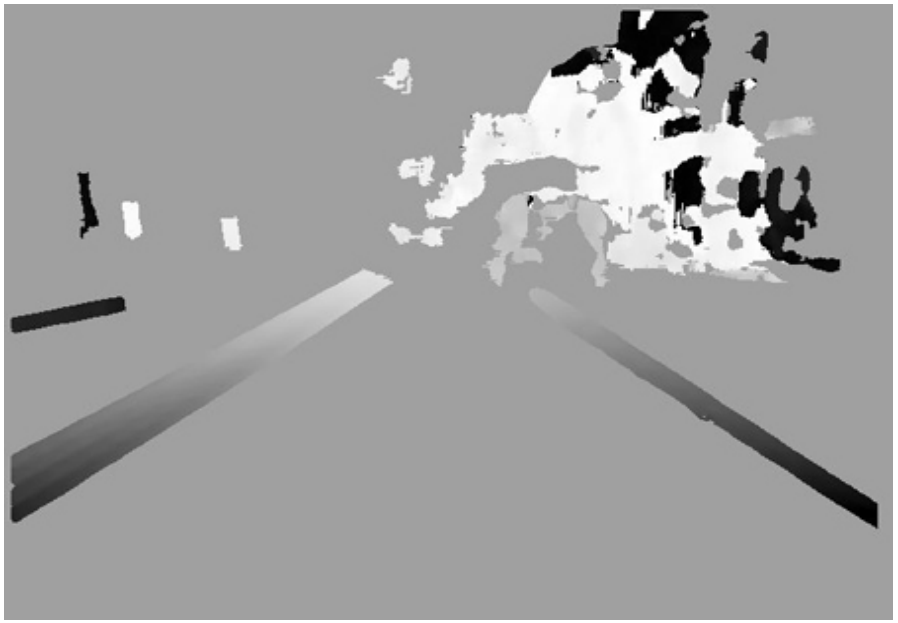
Computer Science Dept.
University of British Columbia
Vancouver, BC, Canada V6T 1Z4

Another Example (Hole Filling)



Cloth Parameters and Motion Capture by David Pritchard B.A.Sc., University of Waterloo, 2001

Depth Map in a City



Stereo Vision

- Large number of algorithms out there:

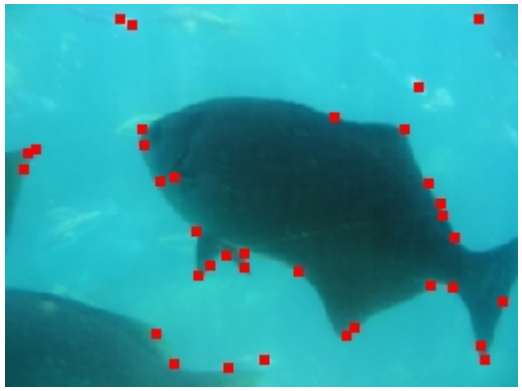
  http://vision.middlebury.edu/stereo/

  rank 43 different algorithms.

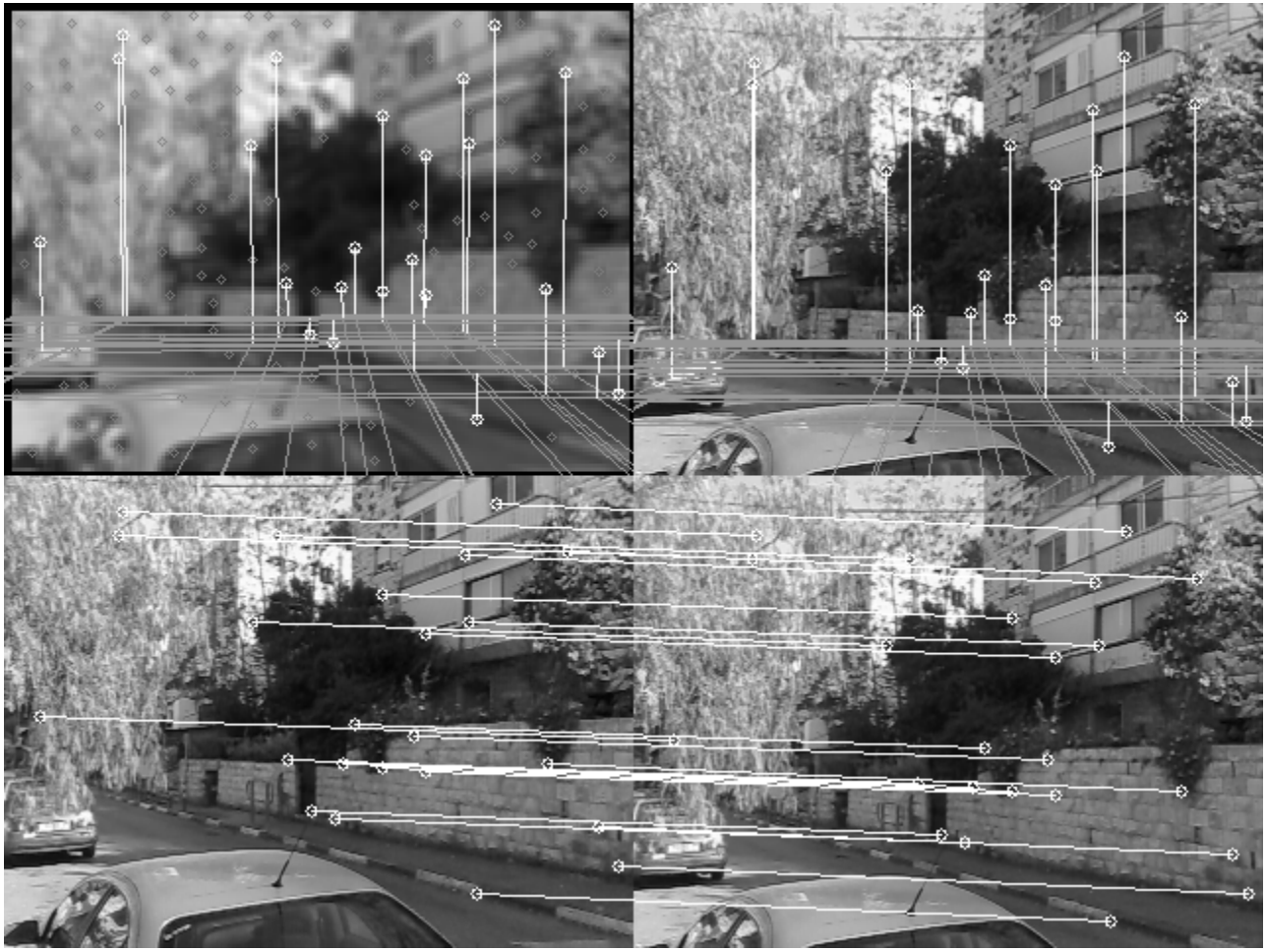
Good Feature

- High Recall
- Good Precision
- Feature Detection
- Feature Matching
- Several Alternatives:
  - Harris Corners (OpenCV)
  - SURF (OpenCV)
  - SIFT
  - Fast
  - etc

Harris Corners



SURF



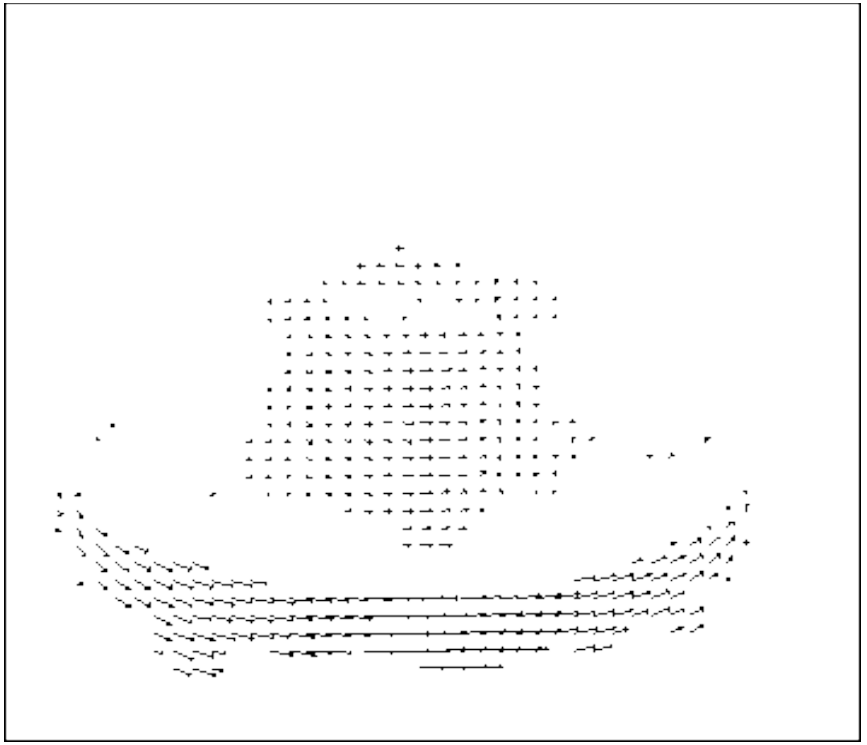
SIFT


BASSONCINI
Grisbì Almond Cookies
Peso Netto 250 g
NET WT 8.8 OZ

Optical Flow

- Definition:
  - *the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene.*

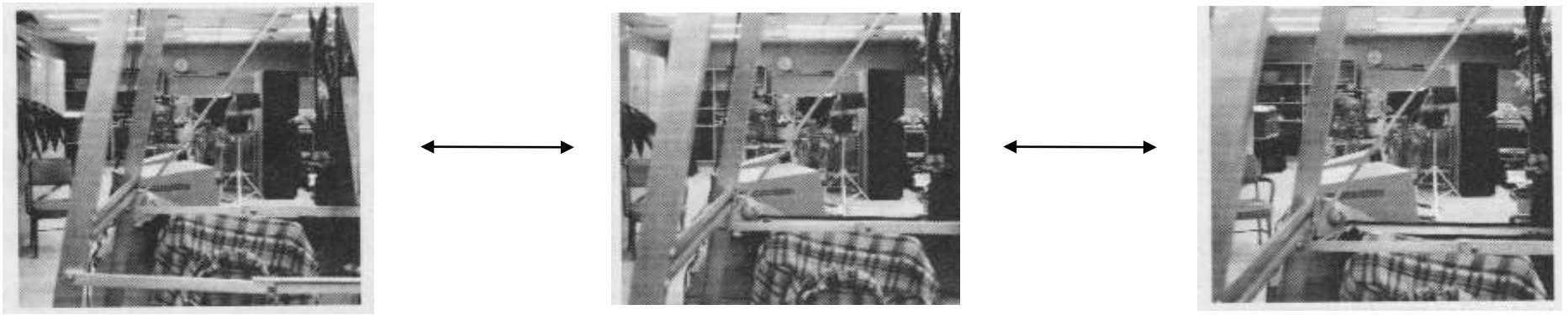
Optical Flow Field



Optical flow

Information about *image motion* rather than the *scene.*
This is a classic ***reconstruction*** *problem.*

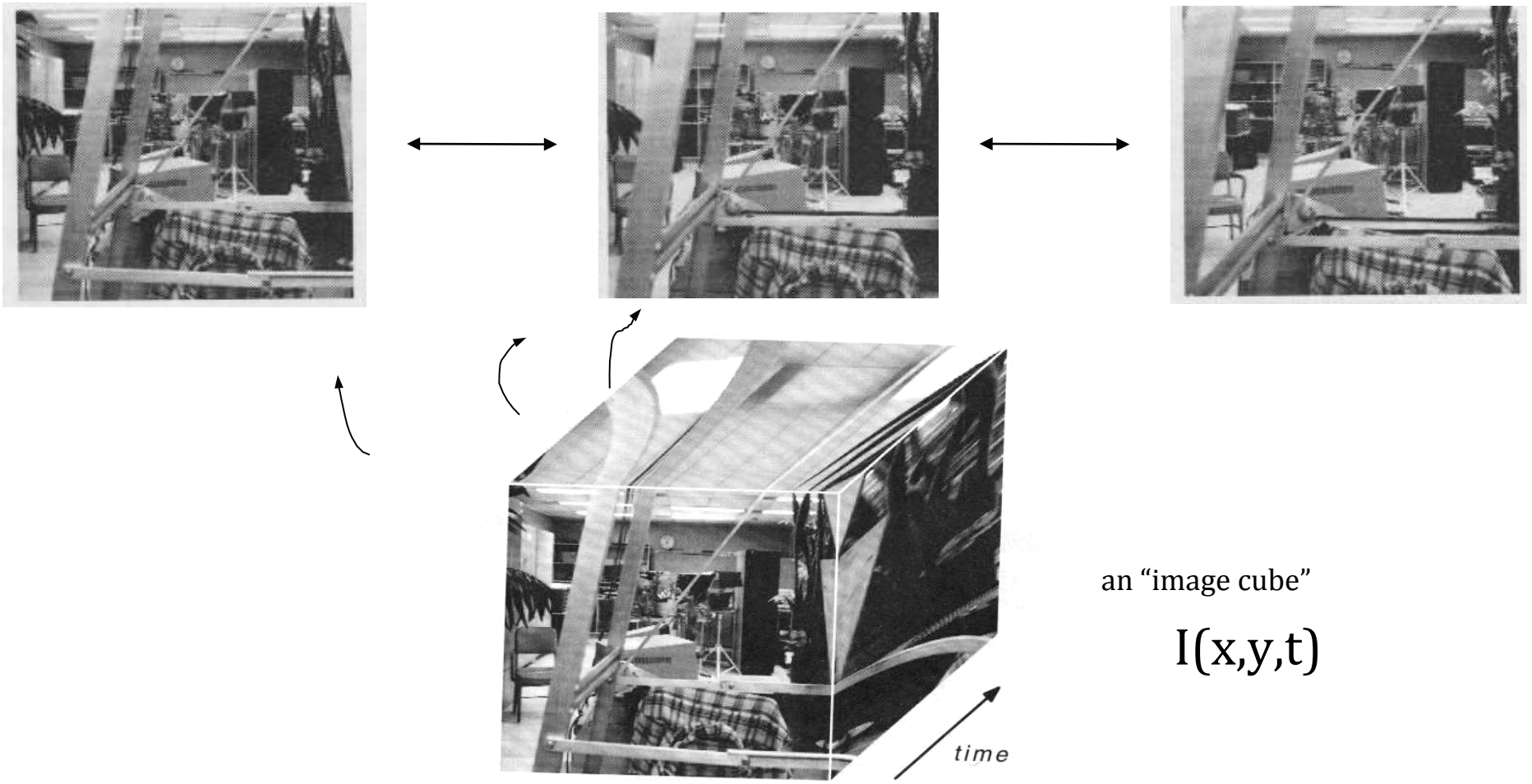
This next step might be to use the image motion to infer scene motion, robot motion or 3D layout.



time sequence of images

Optical flow

Information about *scene motion* rather than the *scene.*

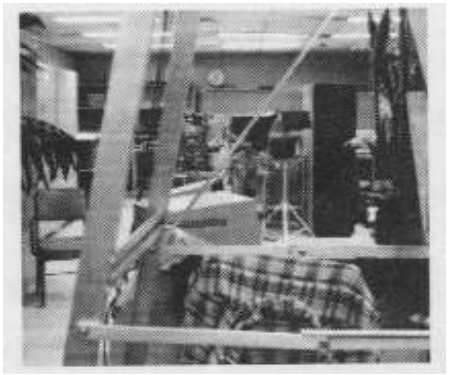


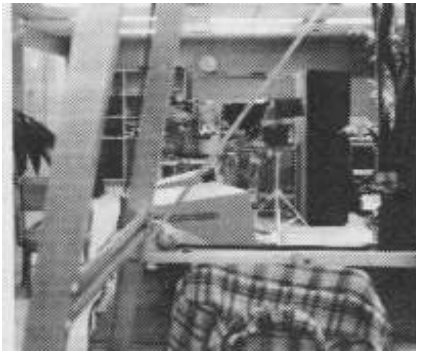
an "image cube"

$I(x,y,t)$

Optical flow

Information about *scene motion* rather than the *scene.*





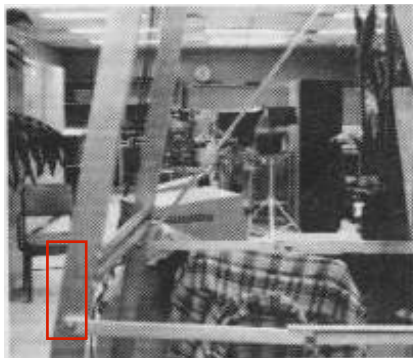
optical flow

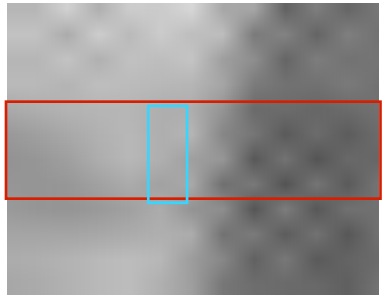
How ?

Optical Flow

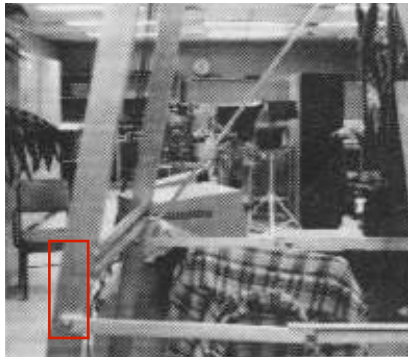
- By measuring the direction that intensities are moving...

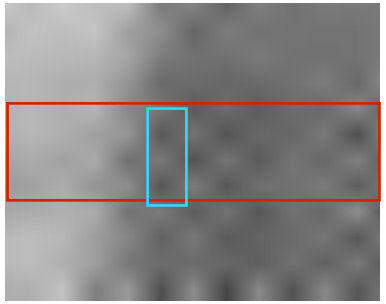
I(x,y,t)





99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	**40**
90	70	50	40	30





90	90	70	40	25
90	70	40	40	25
90	70	**40**	40	25
90	70	40	40	20
70	50	40	30	15

- We can estimate things...

Observations & Warnings

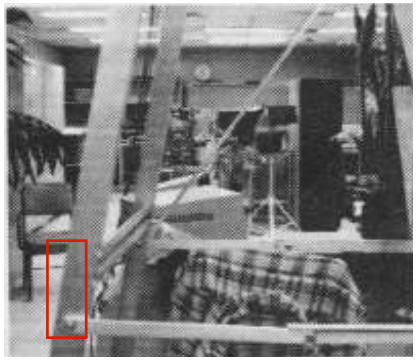
- How can we do this?
- Assume the scene itself is static.
- Find matching chunks in the images.
- An instance of *correspondence*.

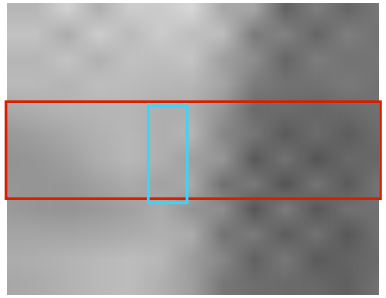
BUT

- World really isn't static.
- Lightning might change even in a static scene.

Optical Flow

By measuring the direction that intensities are moving...





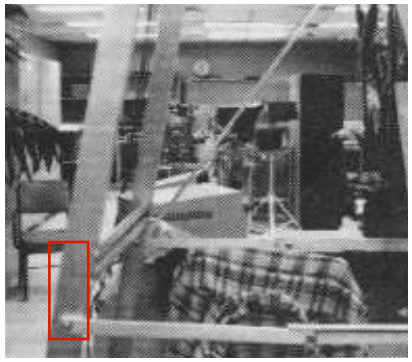
I(x,y,t)

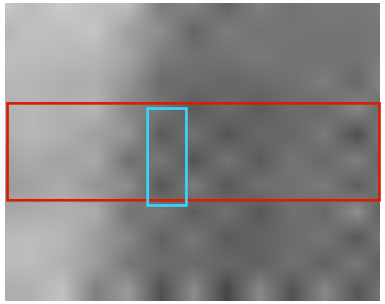
I(x,y,0)

99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	**40**
90	70	50	40	30

I(0,0,0)

I(2,-1,0)





I(x,y,1)

90	90	70	40	25
90	70	40	40	25
90	70	**40**	40	25
90	70	40	40	20
70	50	40	30	15

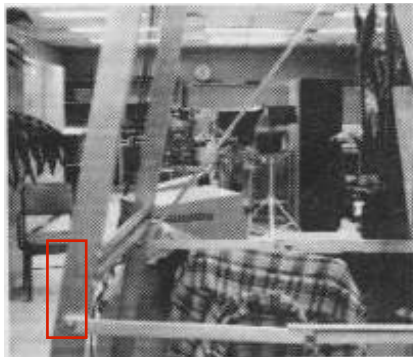
I(0,0,1)

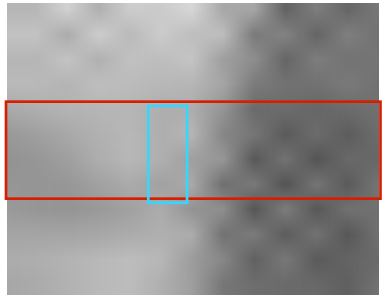
We can estimate things ...

$$\frac{dI}{dx} = I_x \quad \text{at } (0,0,0)$$

Optical Flow

By measuring the direction that intensities are moving...





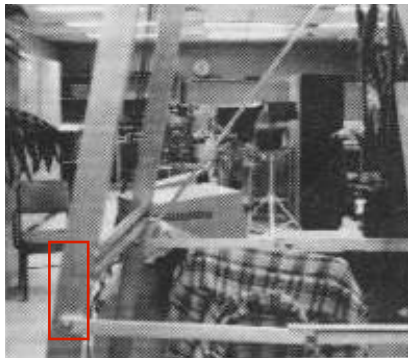
I(x,y,t)

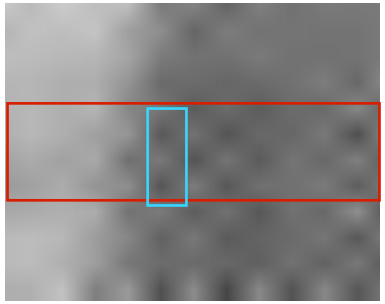
I(x,y,0)

99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	**40**
90	70	50	40	30

I(0,0,0)

I(2,-1,0)





I(x,y,1)

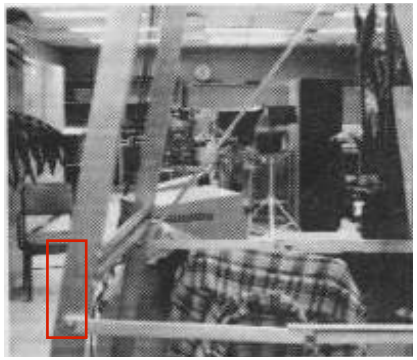
90	90	70	40	25
90	70	40	40	25
90	70	**40**	40	25
90	70	40	40	20
70	50	40	30	15

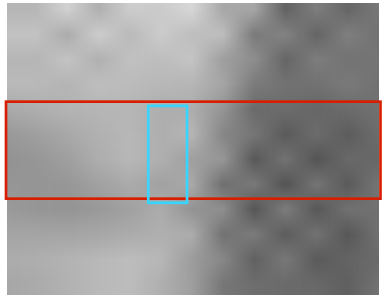
I(0,0,1)

We can estimate things like $\frac{dI}{dx} = I_x$ at (0,0,0) $= \frac{\Delta I}{\Delta x} = \frac{I(1,0,0) - I(0,0,0)}{1 - 0} = -30$

Optical Flow

By measuring the direction that intensities are moving...





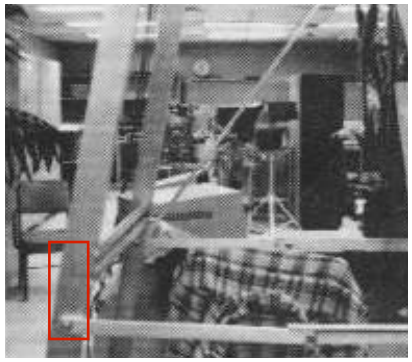
I(x,y,t)

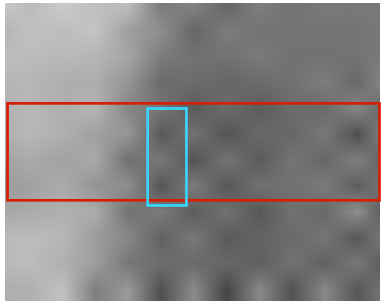
I(x,y,0)

99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	**40**
90	70	50	40	30

I(0,0,0)

I(2,-1,0)





I(x,y,1)

90	90	70	40	25
90	70	40	40	25
90	70	**40**	40	25
90	70	40	40	20
70	50	40	30	15

I(0,0,1)

We can estimate things like

$$\frac{dI}{dx} = I_x \qquad \frac{dI}{dy} = I_y \qquad \frac{dI}{dt} = I_t$$

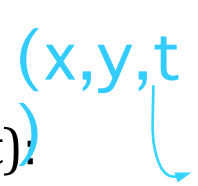
so...

Measuring Optical Flow

Let I(x,y,t) be the sequence of images.

Simplest assumption (constant brightness constraint):

$$I(x,y,t) = I(x + dx, y + dy, t + dt)$$



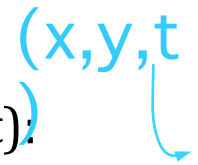
99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	40
90	70	50	40	30

Measuring Optical Flow

Let I(x,y,t) be the sequence of images.

Simplest assumption (constant brightness constraint):

$$I(x,y,t) = I(x + dx, y + dy, t + dt)$$



99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	40
90	70	50	40	30

Reminder: $f(x + dx) = f(x) + f'(x)\,dx + f''(x)\,dx^2/2 + \ldots$

Measuring Optical Flow

Let I(x,y,t) be the sequence of images.

Simplest assumption (constant brightness constraint):

$$I(x,y,t) = I(x + dx, y + dy, t + dt)$$

(x,y,t)

99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	40
90	70	50	40	30

Reminder: $f(x + dx) = f(x) + f'(x)\,dx + f''(x)\,dx^2/2 + \ldots$

$$I(x,y,t) = I(x,y,t) + I_x\,dx + I_y\,dy + I_t\,dt + \text{2nd deriv. + higher}$$

Measuring Optical Flow

Let I(x,y,t) be the sequence of images.

Simplest assumption (constant brightness constraint):

$$I(x,y,t) = I(x + dx, y + dy, t + dt)$$

(x,y,t)

99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	40
90	70	50	40	30

Reminder: $f(x + dx) = f(x) + f'(x)\,dx + f''(x)\,dx^2/2 + \ldots$

$$I(x,y,t) = I(x,y,t) + I_x\,dx + I_y\,dy + I_t\,dt + \text{2nd deriv. + higher}$$

ignore these terms

$$0 = I_x\,dx + I_y\,dy + I_t\,dt$$

Measuring Optical Flow

Let $I(x,y,t)$ be the sequence of images.

(x,y,t)

99	90	90	70	40
95	90	70	40	40
90	90	**70**	40	40
90	90	70	40	40
90	70	50	40	30

Simplest assumption (constant brightness constraint):

$$I(x,y,t) = I(x + dx, y + dy, t + dt)$$

Reminder: $f(x + dx) = f(x) + f'(x)\,dx + f''(x)\,dx^2/2 + \ldots$

$$I(x,y,t) = I(x,y,t) + I_x\,dx + I_y\,dy + I_t\,dt + \text{2nd deriv. + higher}$$

ignore these terms

$$0 = I_x\,dx + I_y\,dy + I_t\,dt$$

$$-I_t = I_x \frac{dx}{dt} + I_y \frac{dy}{dt}$$

intensity-flow equation

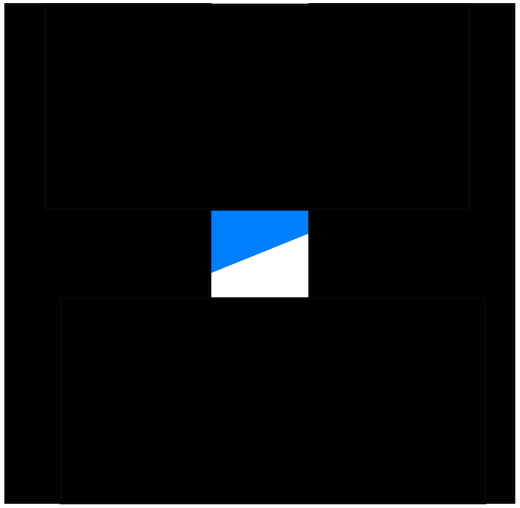
good and bad...

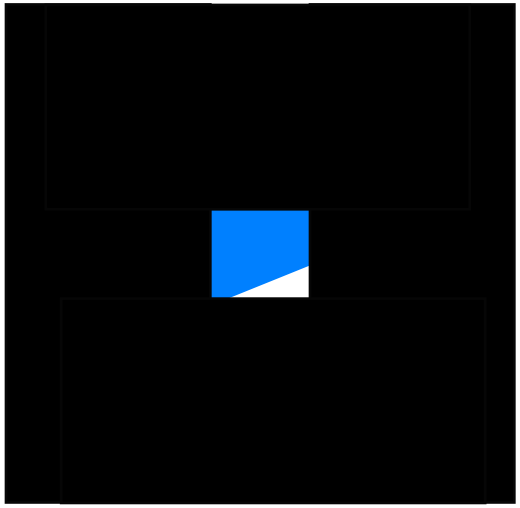
The “aperture” problem

$$-I_t = I_x \frac{dx}{dt} + I_y \frac{dy}{dt}$$

- The intensity-flow equation provides only one constraint on *two* variables (x-motion and y-motion)

→ It is only possible to find optical flow in one direction...





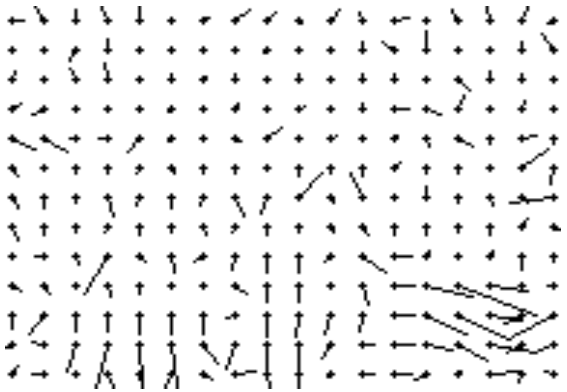
The “aperture” problem

→ It is only possible to find optical flow in one direction...

*at any **single** point in the image !*

img1

img2



raw optical flow

smoothed for ten iterations

Smoothing can be done by incorporating neighboring points’ information.

Optical Flow Application

- Visual Odometry
  - Wheel slip detection on future Mars Rovers

Image Downsampling