Analysis, Buffers and Map Algebra [2020]

GEOG 176B: Technical Issues in GIS / Spatial Data

Krzysztof Janowicz

What You Will Learn in This Lecture

- Understand buffers over raster data an different vector primitives.
- Understand design decisions when selecting buffers, e.g., when to nest buffers or dissolve their boundaries.
- Learn about examples of proximity analysis.
- Review terms such as the scope of operations or moving windows.
- Understand the basics of map algebra and how to apply it to several use cases.

Buffers

Cost Surfaces 000

Some Common Types of Overlays (Recap)

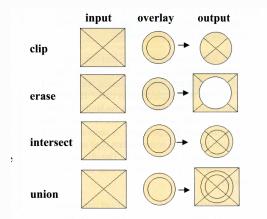


Fig. 8.4. Polygon overlay operations and results. Union and intersect join the attributes of the two input layers. Clip and erase do not.

Cost Surfaces 000

Dissolve Operation (Recap)

Before dissolve





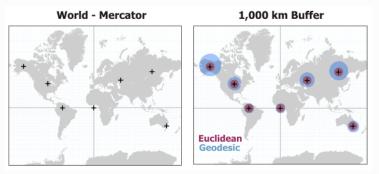
- Creation of aggregated polygons based on merging adjacent polygons that share a common attribute value
- In this neighborhood example, adjacent polygons are merged when they belong to the same class.

Buffers in GIS

- A buffer is a feature or zone surrounding a input feature (or zone) that is generated using some distance (and distance measure).
- This distance can also be measured in time (or other costs).
- Buffers around vector geometries, e.g., points and polygons, are very common, but buffers can also be computed for raster datasets.
- Buffers are areas.
- Buffers can be combined when their borders are adjacent or when the buffers overlap.
- Buffers can be nested based on different distance bins.
- A common example for the usage of buffers is excluding zones, e.g., around a nature preserve.

Cost Surfaces 000

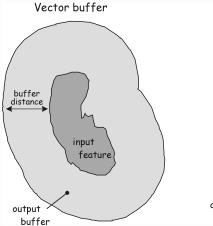
Buffers Distance Methods

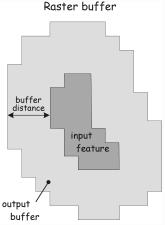


The **Geodesic buffers** (using a geographic coordinate system) the **Euclidean buffers** (using a projected coordinate system) are computed around the same cities; notice how they differ.

Cost Surfaces 000

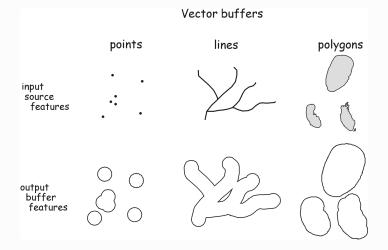
Vector and Raster Buffers





Cost Surfaces 000

Vector Buffers

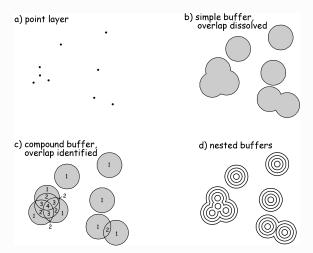


Note how all vector buffers are polygons.

Buffers 0000000000000

Cost Surfaces 000

Point Buffers

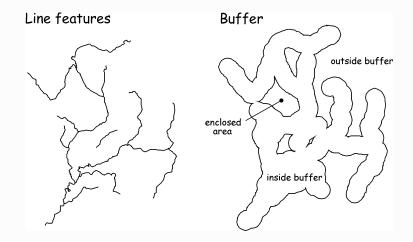


Note how c) translates to densities.

Buffers 000000000000

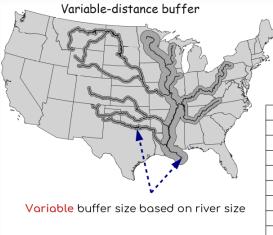
Cost Surfaces 000

Line Buffers



Cost Surfaces 000

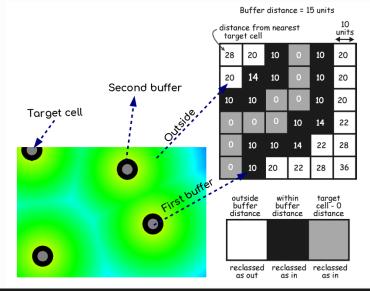
Variable-Distance Buffers



river_identifier	buffdist
mississippi	100
missouri	50
arkansas	50
ohio	75
tennessee	75
st. croix	75
illinois	75
wisconsin	75

Cost Surfaces 000

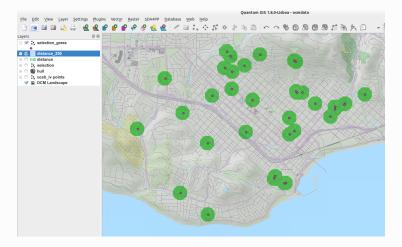
Raster Buffers



Analysis, Buffers and Map Algebra

Cost Surfaces 000

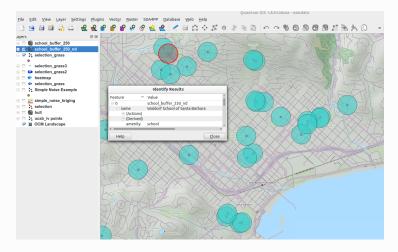
Raster Buffer Example



What can be asked of such raster buffer layer; when to use them?

Cost Surfaces 000

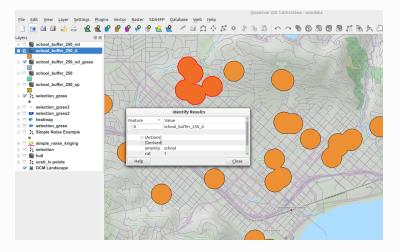
Point-Based Vector Buffer Example



Note how this buffer is about one specific school.

Cost Surfaces 000

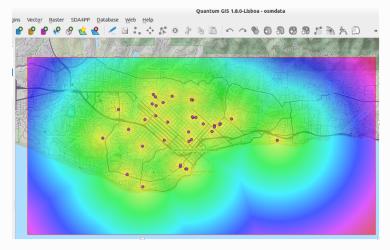
Point-Based Vector Buffer Example



These buffers are **dissolved** using **common** feature **type**.

Cost Surfaces 000

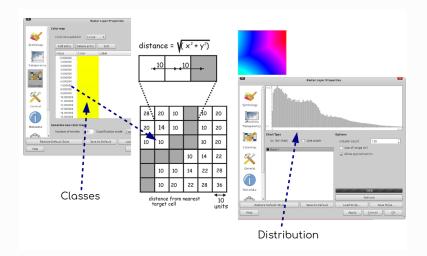
Proximity Map



What do the **colors** mean; what are the **values** of these cells?

Cost Surfaces 000

Proximity Map



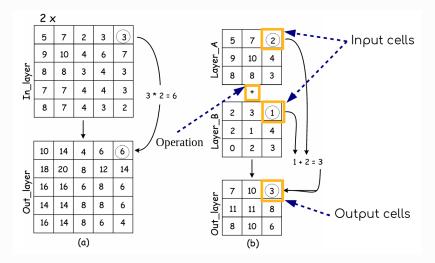
 Cost Surfaces 000

Map Algebra

- Map algebra is a set-based algebra introduced by Tomlin in the 1980s. It describes how to perform (and combine) primitive operations over 1..* geographic (raster) data layers
- These operations often include:
 - Arithmetic operations such as addition or multiplication
 Statistical operations such as means, maxima or minima
 Relational operations such as greater than or (not) equal
 ...
- can be used to combine expressions or perform different operations based on some conditional statement (IF...THEN...)
- Use cases range from everything starting from data cleaning to visualization.

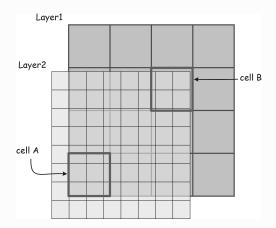
Buffers 00000000000000 Cost Surfaces 000

Map Algebra – Combining Cell Values



 Cost Surfaces 000

Map Algebra – Requirements

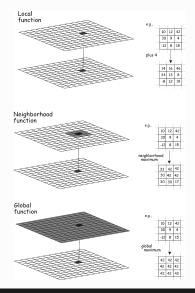


- Before combining cell values, both layers must be geo-registered and of the same spatial resolution.
- You may have to resample the layer.

Scopes of Raster Operations

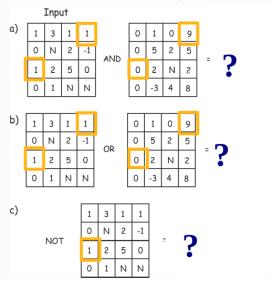
- Similarly, to vector operations raster operations have a local, neighborhood, or global scope.
- Local and neighborhood

scopes are most often used in raster analysis.



Buffers 0000000000000 Cost Surfaces 000

Local Functions – Boolean Operators



Map Algebra

Local Functions – Boolean Operators

		Inp	ut							
a)	1	3	1	1			0	1	0	9
	0	Ν	2	-1	AND		0	5	2	5
	1	2	5	0			0	2	Ν	2
	0	1	N	Ν			0	-3	4	8
b)	1	3	1	1			0	1	0	9
	0	Ν	2	-1	OR		0	5	2	5
	1	2	5	0	0.1		0	2	Ν	2
	0	1	Ν	Ν			0	-3	4	8
								_		
c)				1	3	1	1			0
		NO	-	0	N	2	-1		-	1
		140							-	

N Ν

	0	Juip	uī	
	0	1	0	1
=	0	Ν	1	1
-	0	1	Ν	0

Ν

Ν Ν

NN

Ν

=

Ν

Ν Ν

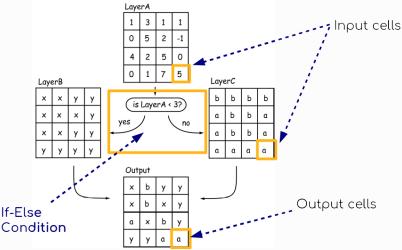
0.....

b)	1	3	1	1	
	0	Ν	2	-1	0
	1	2	5	0	
	0	1	Ν	Ν	

Buffers 000000000000 Cost Surfaces 000

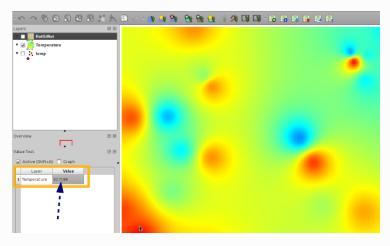
Introducing Conditions/ Control Statements





 Cost Surfaces 000

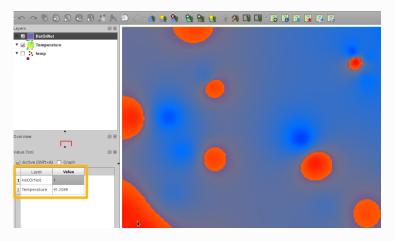
Classify Temperature Example



Based on your personal ranking, is this a very warm/hot temperature spot?

 Cost Surfaces 000

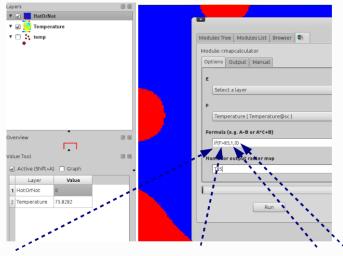
Classify Temperature Example



How to create a raster map showing hot areas based on your ranking?

 Cost Surfaces 000

Classify Temperature Example

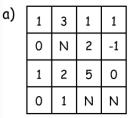


If a cell temperature value is greater than 85, set the cell value to 1, else 0.

Buffers 000000000000 Cost Surfaces

Reclassification Operations







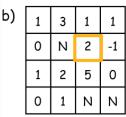
in	out	
0	۵	
1	x	
2	b	
3	f	
4	с	
5	s	

=

=

	P		
x	f	x	x
۵	Ν	b	Z
x	b	S	۵
۵	x	Ν	Ν

Output



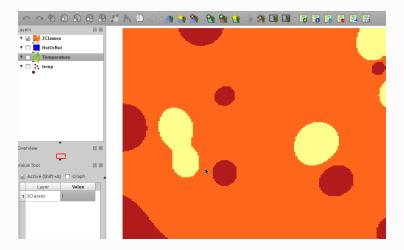


in range	out
0 to 1.5	۵
1.5 to 3.5	b
3.5 to 10	с
Ν	d

۵	b	۵	۵
۵	d	b	Ν
۵	b	с	۵
۵	b	d	d

 Cost Surfaces 000

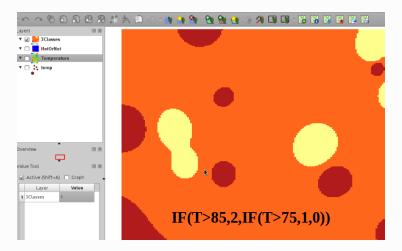
Classify Temperature Example with 3 Classes



How to create a raster map showing hot, warm, and cold areas?

 Cost Surfaces 000

Classify Temperature Example with 3 Classes

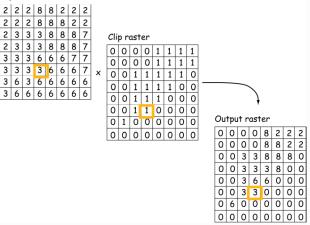


How to create a raster map showing hot, warm, and cold areas?

Buffers 0000000000000 Cost Surfaces 000

The Map Algebra Realization of a Clip Overlay

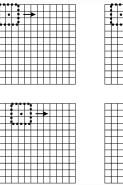
Input raster

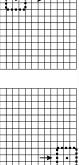


How to use just one operator to realize an **erase overlay**?

 Cost Surfaces 000

Neighborhood Functions – Moving Window

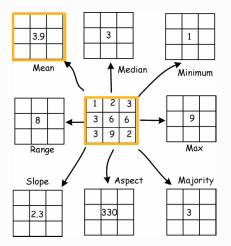




- The value for the cell in the center of each window is computed by all the 3x3 neighbor cells.
- The window moves cell by cell for each row and column
- Why 3x3 or 5x5 cells but not 4x4 cells?

 Cost Surfaces 000

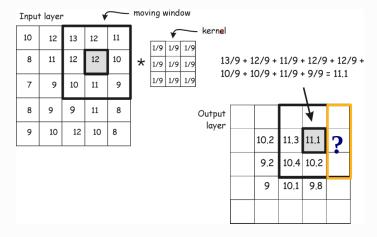
Moving Windows and Neighborhood Functions



How would you realize the **mean** using map algebra?

 Cost Surfaces 000

Kernels

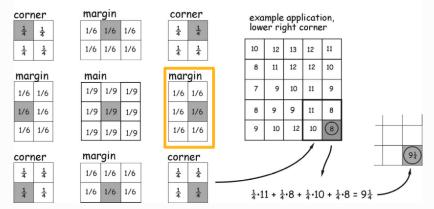


A **kernel** is the set of **constants** for the cell values in a given **window**.

 Cost Surfaces 000

Different Kernels for Corners and Margins

Mean function kernels

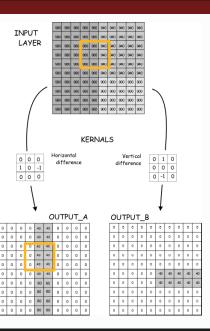


Select a different kernel window for corners or margins or a **larger study area**.

Buffers 00000000000000 Cost Surfaces 000

Kernel-Based Edge Detection

- Detect abrupt changes in cell values using a kernel.
- Example:



Buffers 00000000000000

Kernel-based High-Pass Filter

100	100	100
100	100	100
100	100	100

(-800+900) / 9 = 11

100	100	100				
100	1000	100				
100	100	100				

(-800+9000) / 9 = 911

	I	nput	t La	yer	wi	th	"no	ise"								
~	1065	1058	929	864	960	1113	974	896	890	841	759	719	705	696	720	708
·	1038	963	947	960	999	1021	1011	1015	995	1044	870	773	734	703	676	684
kernal for high-pass filter	1142	1005	1151	310	1117	1056	1007	1002	902	854	935	913	789	756	724	700
-1 -1 -1	1116	1114	1270	1165	1097	1025	922	917	821	829	860	838	807	810	758	760
$\frac{1}{9} \times \frac{-1}{-1} \frac{-1}{-1} \frac{-1}{-1}$	1275	1170	1295	1114	1009	942	963	847	835	729	738	797	723	718	694	670
/	1441	1263	1196	1055	913	869	829	771	736	766	766	688	694	1676	684	698
	1348	2900	1096	909	948	951	940	897	818	953	784	732	704	733	776	804
(1377	1238	1122	1019	1089	850	964	896	2000	800	760	666	779	867	896	744
1	1489	1320	1188	1152	1050	942	922	952	815	841	721	780	852	28	845	738
(-1)*1065 + \	1432	1415	1196	1100	1001	<u>974</u>	924	911	914	758	809	861	898	830	748	710
(-1)*1068 +	1412	1474	1240	1100	1001	982	873	835	829	853	931	937	845	708	685	680
(-1)*924 + (-1)*1038 +	1493	1358	1201	1090	1054	970	902	902	958	952	1015	841	782	803	786	711
(9)*963)+ (-1)*947 +	1437	1407	118	1145	1070	1107	982	1047	1077	1062	954	884	44	940	828	771
(-1)*1142 +	1349	1359	1267	1247	1194	1196	1077	1214	1145	999	906	894	1024	1046	923	862
(-1)*1005 + (-1)*1151	1319	1292	1378	1400	1367	1276	1162	1085	961	930	872	985	1010	1178	1145	1000
9 / /						sp	ikes				1	pits				
1	C	upu	tL	aye	r	sp	ikes			1	'	pits				
= 35.7		Ĺ		Ĺ					450	250			64	60	17	
1	, ,		142	141	177	105	112	162	156	250	99	59	01	50	43	
1		Ĺ		Ĺ					156 60	250 147 76			01 85	50 89	43 74 110	
1	, ,	33	142	141	177 270	106 145	112	102	60	147	99 148	59 179	85	89	74	
1	,	33	142 292 345	141 -657 235	177 270 237	106 145 124	112 121 61	1G2 158 107	60 41	147 76	99 148 111	59 179 108	85 102	89 146	74 110	
1		33 33 68 62	142 292 345 258	141 -657 235 114	177 270 237 100	105 145 124 85	112 121 61 161	102 158 107 93	60 41 122	147 76 23	99 148 111 41	59 179 108 117	85 102 -58	89 146 -43	74 110 -59	
1	,	35 33 88 62 -35	142 292 345 258 -7	141 -657 235 114 110	177 270 237 100 40	105 145 124 95 37	112 121 61 161 36	102 158 107 93 12	60 41 122 14	147 76 23 68	99 148 111 41 88	59 179 108 117 28	85 102 -58 -59	89 146 -43 1099	74 110 -59 -69	
1	o J	33 33 66 62 -35	142 292 345 258 -7 -140	141 -057 235 114 110 35	177 270 237 100 40 79	105 145 124 95 37 118	112 121 61 161 35 152	102 158 107 93 12 -16	60 41 122 14 -38	147 76 23 68 37	99 148 111 41 88 109	59 179 108 117 28 79	85 102 -58 -59 -59	89 146 -43 1000 -54	74 110 -59 -69 -14	
1		35 33 66 62 -35 1784 -74	142 292 345 258 -7 -140 -83	141 -657 235 114 110 35 66	177 270 237 100 40 79 202	105 145 124 85 37 118 83	112 121 161 161 152 131	162 158 107 93 12 -16 -23	60 41 122 14 -38 1995	147 76 23 68 37 -45	99 148 111 41 88 109 69	59 179 108 117 28 79 18	85 102 -58 -59 -59 179	89 146 -43 1999 -54 243	74 110 -59 -69 -14 281	
1		35 33 66 62 -35 1704 -74	142 292 345 -7 -140 -83 125	141 -657 235 114 110 35 66 178	177 270 237 100 40 79 202 135	105 145 124 95 37 118 83 68	112 121 61 161 152 131 68	102 158 107 93 12 -16 -23 25	60 41 122 14 -38 1225 -82	147 76 23 68 37 -45 -1	99 148 111 41 88 109 69 20	59 179 108 117 28 79 18 71	85 102 -58 -59 179 214	89 146 -43 1000 -54 243 -718	74 110 -59 -69 -14 281 227	
1		35 33 66 62 -35 1784 -74 158 220	142 292 345 256 -7 -140 -83 125 96	141 -657 235 114 110 35 66 178 108	177 270 237 100 40 79 202 135 78	105 145 124 95 37 118 83 68 119	112 121 61 161 152 131 68 102	102 158 107 93 12 -18 -23 25 126	60 41 122 14 -38 1245 -82 159	147 76 23 68 37 -45 -1 10	99 148 111 41 88 109 69 20 96	59 179 108 117 28 79 18 71 108	85 102 -58 -59 -59 179 214 249	89 146 -43 1039 -54 243 -718 207	74 110 -59 -14 281 227 165	
1		35 33 66 62 -35 7784 158 220 278	142 292 345 256 -7 -140 -83 125 99 135	141 -057 235 114 110 35 66 178 108 111 207	177 270 237 100 40 79 202 135 78 80	105 145 124 95 37 118 83 68 119 125	112 121 61 161 152 131 68 102 50	162 158 107 93 12 -16 -23 25 125 33	60 41 122 14 -38 -62 159 42	147 76 23 68 37 -45 -1 10 57	99 148 111 41 88 109 89 20 96 150	59 179 108 117 28 79 18 71 108 101	85 102 -58 -59 -59 179 214 249 105	89 146 -43 1002 -54 243 -710 207 -3	74 110 -59 -69 -14 281 227 165 21	
1		35 33 66 62 -35 7104 158 220 278 281	142 292 345 -7 -140 -83 125 96 135 207	141 -057 235 114 110 35 66 178 108 111 207	177 270 237 100 40 79 202 135 78 80 123	105 145 124 95 37 118 83 68 119 125 83	112 121 61 161 161 152 131 68 102 50 46	162 158 107 93 12 -16 -23 25 125 33 68	60 41 122 14 -38 -62 159 42 119	147 78 23 68 37 -45 -1 10 57 99	99 148 111 41 88 109 89 20 66 150 192	59 178 108 117 28 79 18 71 108 161 130	85 102 -58 -59 179 214 249 105 115	89 148 -43 -54 243 -718 207 -3 179	74 110 -59 -69 -14 281 227 165 21 105	

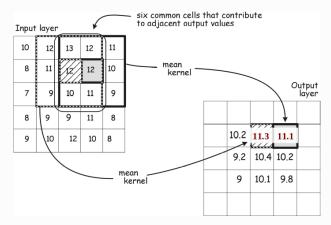
 Cost Surfaces

Using an Averaging Kernel for Smoothing

	òo	D		8	$_{\sigma K}^{20}$	at a	۵	° 📣	2	e	2	1		§ \$	🦋 (V		- 💽	3 💽	8 🔀
_ayers				×									•							
🔻 🖌 🛃 3Cla	sesAve	rageKe	ernel																	
🔻 🖌 🔛 3Cla	ses												M	odule	es Tre	e I	٥dul	es List	Bro	wser
🔻 🗌 📴 HotO	rNot												N	/lodul	e: r.n	eighl	bors			
🔻 🖌 🚺 Tem	perature													Optio	ons	Out	put	Manu	al	
▼ □ 🚼 temp														Nam	ie of i	inpu	t rast	erma	p	
													Ш	:	Class	ses (3Clas	ses@s	c)	
												2		Neig	ghbor	rhoo	d ope	ration		
													Ш	/	Avera	ge				
														Neig	ghbor	rhoo	d size			
													Ш	3	3					
Overview	-		Ø	×		-				ß				Nam	e for	out	put ra	ister r	nap	
/alue Tool	•••		Ø	×										3	Class	sesA	verag	eKern	el	
🖌 Active (Shif	:+A) 🗆	Graph	n	-									l							
Layer		Value															17			
1 3ClassesAve	0																		Run	
2 3Classes	1																		Kurl	

 Cost Surfaces 000

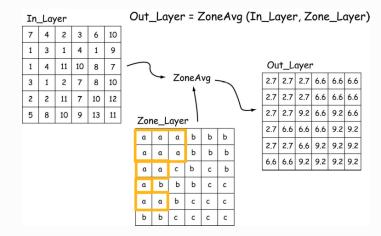
Kernel Functions and Spatial Covariance



Keep in mind that kernel functions will **increase** the **spatial covariance** of cell values. For adjacent cells, **6 out of 9 cells** used to compute the new value will be the same.

Map Algebra 0000000000000000000000000 Cost Surfaces 000

Zonal Functions

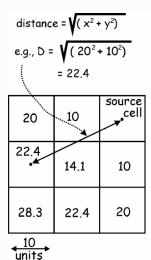


The term **neighborhood** may also be **generalized** for **regions** or zones.

Buffers 0000000000000

Cost Surfaces ●00

Cost Surfaces As Analogy to Network Analysis



cost = distance * fixed cost factor e.g., cost = distance * 2

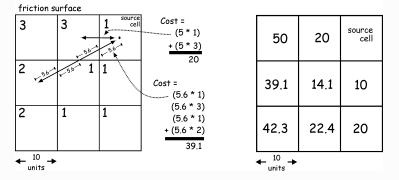
40	20	source cell
44.8	28.2	20
56.6	44.8	40

Cost Surfaces 0●0

Cost Surfaces As Analogy to Network Analysis

cost = cell distance * friction

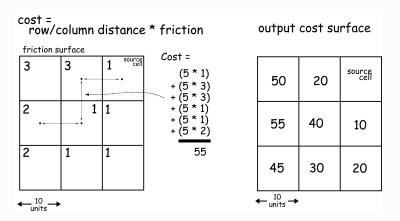
output cost surface



Instead of a cost constant, **costs** can be modeled as **cell values**.

Cost Surfaces

Row-Column Distance



All edges have the same length – either 5 or 10 units.