

Master Degree in Computer Science

# Scientific and Large Data Visualization

*Francesco Caprari*  
Academic Year: 2023-2024

# Contents

<b>1</b>	<b>Intro</b>	<b>6</b>
1.1	Information Visualization . . . . .	6
1.1.1	Perchè la visualizzazione? . . . . .	7
1.2	Fundamentals of 3D Computer Graphics . . . . .	8
1.3	Scientific Visualization . . . . .	9
<b>2</b>	<b>Information Visualization</b>	<b>10</b>
2.1	Data . . . . .	10
2.2	Tipi di attributi . . . . .	10
2.3	Semantica degli attributi . . . . .	11
2.4	Grafici Fondamentali (Bivariate data) . . . . .	11
2.4.1	Bar Charts . . . . .	11
2.4.2	Histograms . . . . .	12
2.4.3	Pie Charts . . . . .	12
2.4.4	Donut Charts . . . . .	13
2.4.5	Scatter plots . . . . .	14
2.4.6	Slope Charts . . . . .	14
2.4.7	Line Charts . . . . .	15
2.4.8	Area Charts . . . . .	15
2.4.9	Chropleth maps . . . . .	15
2.4.10	Symbol maps . . . . .	16
2.5	Grafici Fondamentali (Attributi Multipli) . . . . .	16
2.5.1	Stacked Bar Charts . . . . .	16
2.5.2	Grouped Bar Charts . . . . .	17
2.5.3	Stacked Line Charts . . . . .	17
2.5.4	Line Charts Series . . . . .	18
2.5.5	Bubble Charts . . . . .	18
2.5.6	Small multiples . . . . .	19
2.5.7	Symbol Maps . . . . .	19
2.6	Grafici Fondamentali (Attributi Gerarchici) . . . . .	20
2.6.1	Sunburst Charts . . . . .	20
2.6.2	Treempas . . . . .	20
<b>3</b>	<b>Information Visualization II</b>	<b>22</b>
3.1	Visual Encoding . . . . .	22
3.2	Graphical Elements . . . . .	22
3.3	Visual decoding . . . . .	23
3.3.1	Quality Evaluation . . . . .	23
3.3.2	Percentuale di Accuracy . . . . .	24
3.4	Altri tipi di grafici . . . . .	24
3.4.1	WordClouds . . . . .	24
3.4.2	Calligrams . . . . .	25
3.4.3	Flow Maps . . . . .	25
3.4.4	Chernoff faces . . . . .	26
3.4.5	Multidimensional Icons . . . . .	26

3.4.6	Petal as a gliph . . . . .	27
<b>4</b>	<b>Applied Perception</b>	<b>28</b>
4.1	Illusions . . . . .	28
4.1.1	Mach Banding . . . . .	28
4.1.2	Hermann Grid illusions . . . . .	29
4.1.3	Chevreul Illusion . . . . .	30
4.1.4	Greyscale Maps . . . . .	30
4.1.5	The Cornsweet Effect . . . . .	30
4.2	Eye Movements . . . . .	31
4.2.1	Preattentive Processes . . . . .	31
4.2.2	Gestalt Laws . . . . .	32
4.3	Illusions . . . . .	33
4.3.1	Muller-Lyer . . . . .	33
4.3.2	Wundt . . . . .	33
4.3.3	Hering . . . . .	34
4.3.4	Horizontal–Vertical Illusion . . . . .	35
4.3.5	Comparing Area . . . . .	35
4.3.6	Weber’s Law . . . . .	36
<b>5</b>	<b>Applied Perception II</b>	<b>37</b>
5.1	Color Vision . . . . .	37
5.2	Trichromacy and Opponent process theories . . . . .	37
5.2.1	Trichromacy theory . . . . .	37
5.2.2	colour measurement and specifcation . . . . .	38
5.2.3	CIE XYZ Color Space . . . . .	38
5.2.4	Opponent process theory . . . . .	39
5.3	Color Space . . . . .	40
5.3.1	RGB Color Space . . . . .	40
5.3.2	HSV/HSL color space . . . . .	40
5.3.3	CIE Lab/Lch color space . . . . .	40
5.3.4	Color differences . . . . .	41
5.4	Color and visualization . . . . .	41
5.4.1	Luminance and visualization . . . . .	41
5.4.2	Saturation and visualization . . . . .	41
5.4.3	Color for Labeling . . . . .	42
5.4.4	Color and semantics . . . . .	42
5.4.5	Color Ramps . . . . .	42
<b>6</b>	<b>MultidimensionalData</b>	<b>43</b>
6.1	Multi-Set Bar Charts . . . . .	43
6.2	Multiple Bars . . . . .	43
6.3	Stacked Bar Charts . . . . .	44
6.4	SpinePlots . . . . .	44
6.5	Mosaic Plots . . . . .	45
6.6	Tree Maps . . . . .	45
6.7	Parallel Coordinates . . . . .	46

6.8	Star Plot . . . . .	46
<b>7</b>	<b>Dimensionality Reduction</b>	<b>47</b>
7.1	Principal Component Analysis (PCA) . . . . .	47
7.1.1	Signal-to-noise Ratio (SNR) . . . . .	47
7.2	Covariance Matrix & Risolvere PCA . . . . .	48
7.2.1	limit of PCA . . . . .	48
7.2.2	PCA and MDS . . . . .	48
7.3	Locally Linear Embedding . . . . .	48
7.4	ISOMAP . . . . .	49
7.5	Autoencoders . . . . .	49
7.6	SNE, Stochastic Neighbor Embedding . . . . .	50
<b>8</b>	<b>3D Data Representation</b>	<b>51</b>
8.1	Gemoetry Processing . . . . .	51
8.1.1	Surface in Computer Graphics . . . . .	51
8.1.2	Voxels . . . . .	52
8.1.3	Point Clouds . . . . .	53
8.1.4	Polygonal meshes . . . . .	54
8.1.5	Simplicies . . . . .	55
8.1.6	Simplicies complexes . . . . .	56
8.2	Star and Link . . . . .	57
8.2.1	Surface in Computer Graphics . . . . .	57
8.2.2	From continuos to discrete setting . . . . .	58
8.3	Orientabilità . . . . .	58
8.3.1	Tangent and normal vectorscon curves . . . . .	58
8.3.2	Normals on triangle meshes . . . . .	59
8.4	Curvature . . . . .	59
8.4.1	Normal curvature on surfaces . . . . .	59
8.4.2	Mean and Gaussian curvature . . . . .	60
8.5	Geodesics . . . . .	60
8.6	Genus . . . . .	61
<b>9</b>	<b>Geometric Transformations</b>	<b>63</b>
9.1	Recap: Point and Vectors . . . . .	63
9.2	Affine transformations 2D . . . . .	64
9.2.1	Translation(2D) . . . . .	64
9.2.2	Scaling(2D) . . . . .	64
9.2.3	Rotation(2D) . . . . .	65
9.2.4	Composing Transformations . . . . .	65
9.2.5	Homogeneous coordinates (2D) . . . . .	65
9.2.6	Translation revisited (2D) . . . . .	66
9.2.7	Scaling revisited (2D) . . . . .	66
9.2.8	Scaling Revisited . . . . .	66
9.2.9	Rotation revisited (2D) . . . . .	67
9.2.10	Riassumendo . . . . .	67
9.2.11	Concatenation of transformations . . . . .	67

9.2.12	Reflection (2D) . . . . .	68
9.2.13	Shear (2D) . . . . .	68
9.3	Affine transformations 3D . . . . .	69
9.3.1	Rotation(3D) . . . . .	69
9.3.2	Recap from the rendering pipeline . . . . .	69
9.3.3	Prospective and Parallel projections . . . . .	70
9.3.4	Perspective projection matrix . . . . .	71
9.3.5	Parallel projection matrices . . . . .	71
<b>10</b>	<b>Introduction to Rendering</b>	<b>73</b>
10.1	Pinhole Camera . . . . .	73
10.1.1	Perspective Projection . . . . .	74
10.1.2	From image plane to pixel coordinates . . . . .	74
10.1.3	Camera Model . . . . .	74
10.2	Ray-Tracing . . . . .	75
10.3	Rasterization-Based Rendering . . . . .	77
10.4	( . . . . .	78
10.4.1	Visione Storica . . . . .	79
10.4.2	Reality . . . . .	79
<b>11</b>	<b>Texturing</b>	<b>80</b>
11.1	Texels . . . . .	80
11.2	Attribute Interpolation & Coordinare Baricentriche . . . . .	81
11.2.1	interpolazione delle coordinate della texture . . . . .	81
11.3	UV Mapping . . . . .	82
11.3.1	Enviroment mapping: Sphere mapping . . . . .	83
11.3.2	Enviroment mapping: Cube mapping . . . . .	83
11.3.3	Texturing out of range . . . . .	84
11.4	Texture Look-Up . . . . .	84
11.4.1	Maginification . . . . .	85
11.4.2	Minification . . . . .	85
11.4.3	Mip-Mapping . . . . .	85
11.4.4	Bump Mapping . . . . .	85
11.4.5	Displacement Mapping . . . . .	86
<b>12</b>	<b>Lighting</b>	<b>87</b>
12.1	Modelli Matematici della luce . . . . .	87
12.2	Solid angle . . . . .	87
12.3	Radiometry . . . . .	88
12.4	e BSSRDF (Bidirectional Scattering Surface Reflectance Distribution Function) . . . . .	88
12.5	The Rendering Equation . . . . .	88
12.5.1	Local and Global effects . . . . .	89
12.5.2	Phong Lighting model . . . . .	89
12.5.3	Fresnel Law . . . . .	90
12.5.4	Lambert Law . . . . .	90
12.5.5	Diffuse Reflection . . . . .	90

12.5.6 Specular Reflection . . . . .	90
12.5.7 Ambient Component . . . . .	91
12.5.8 Shading . . . . .	91
12.6 Gouraud Shading . . . . .	92
12.6.1 Phong Shading . . . . .	93
<b>13 Scientific Visualizzazione</b>	<b>94</b>
13.1 Basi di Scientific Visualization . . . . .	95
13.1.1 Domain . . . . .	95
13.1.2 Attributi . . . . .	96
13.1.3 Transfer functions . . . . .	96
13.1.4 Volume Rendering . . . . .	97
13.1.5 Critical Points, continuous Case . . . . .	97
13.1.6 Morse functions . . . . .	98
13.1.7 Pice-wise linera (PL) settings . . . . .	98
13.1.8 Filtration . . . . .	99
13.1.9 Diagramma di Persistenza . . . . .	100
13.1.10 Reeb graph . . . . .	101
13.2 Vector field visualization . . . . .	102
13.2.1 Tensor and Vector field visualization . . . . .	103
13.3 Scientific illustration . . . . .	104

# 1 Intro

## Information Visualization

- Quando la rappresentazione visiva non è ovvia...
- Percezione visiva, migliori pratiche, dati multidimensionali, grafi e reti

## 3D Computer Graphics

- Rappresentazione in 3D (mesh poligonali)
- Trasformazione dei dati 3D in immagini generate al computer: rendering, illuminazione, texturizzazione

## Scientific Visualization

- Illustrazione grafica dei dati scientifici per estrarre informazioni sui fenomeni

### 1.1 Information Visualization

Consiste nella Trasformazione dei dati per estrarre informazioni utili, ci sono diverse definizioni:

- La visualizzazione delle informazioni è l'uso di rappresentazioni visive, computerizzate, interattive, di dati astratti per amplificare la cognizione [S. T. Card, 1999].
- Visualizzare significa rendere visibili e comprensibili certi fenomeni e parti della realtà; molti di questi fenomeni non sono naturalmente accessibili all'occhio nudo, e molti di essi non sono neanche di natura visiva [J. Costa, 1998].
- Visualizzazione delle informazioni vs visualizzazione scientifica: Nella visualizzazione scientifica esiste una relazione naturale tra ciò che viene rappresentato e la sua rappresentazione, mentre nella visualizzazione delle informazioni la relazione è convenzionale.

Ci sono due prospettive diverse nell'interpretare l'Information Visualization:

- **Visualizzazione come scienza applicata:** Il valore di una buona visualizzazione consiste nel permetterci di individuare pattern nei dati e quindi la scienza della percezione dei pattern può fornire una base per decisioni di design [C. Ware, Information Visualization. Perception for design, 2013].
- **Visualizzazione come tecnologia:** Gli infografici sono uno strumento visivo per la comunicazione, la comprensione e l'analisi [A. Cairo, The functional art, 2013]. Le limitazioni funzionali formano: come il design di un oggetto tecnologico deve dipendere dal compito che dovrebbe aiutare la forma grafica dovrebbe essere vincolata dalle funzioni della tua presentazione.

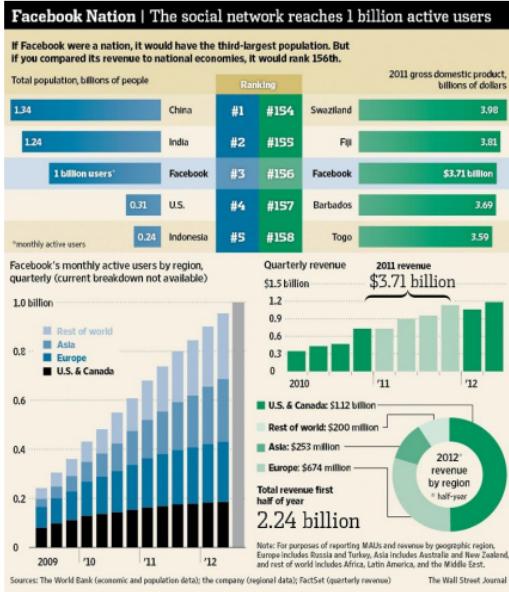


Figure 1: Visualizzazione dati di Netflix

### 1.1.1 Perchè la visualizzazione?

Le visualizzazioni esplicative sono strumenti per presentare informazioni, comunicare dati e messaggi, spiegare qualcosa a qualcun altro, ha tre scopi principali: Explanation, confirmation e exploration.

Siamo una specie visiva. Il sistema visivo umano è molto bravo nell'identificare e analizzare i pattern. Le visualizzazioni sono come artefatti cognitivi (strumenti che gli esseri umani hanno costruito per aiutare a pensare meglio, ad esempio l'abaco). La visualizzazione ha a che fare con la cognizione distribuita (il nostro sistema cognitivo non è esclusivamente composto dal nostro cervello, mente e sensori, ma anche dall'ambiente intorno a noi, che utilizziamo per memorizzare e manipolare informazioni). Le visualizzazioni permettono il trattamento parallelo delle informazioni, anziché sequenziale. Le visualizzazioni esplorative sono strumenti per i lettori per analizzare ciò che viene loro presentato. Molto spesso, l'esito dell'analisi esplorativa non è solo la risposta alle domande originali, ma la generazione di nuove domande.

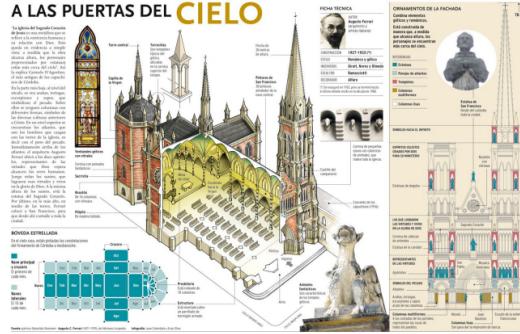


Figure 2: Juan Colombata & Enzo Oliva La Voz del Interior

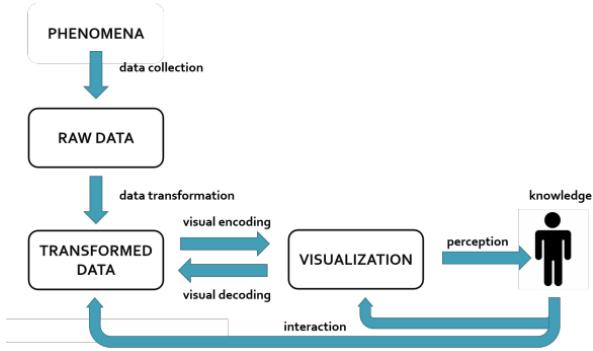


Figure 3: Visualization pipeline

## 1.2 Fundamentals of 3D Computer Graphics

Sono molti i campi dove si utilizza la **modellazione 3D**: Animazioni e serious gaming, Progettazione assistita da computer (CAD) e modellazione di prodotti, Patrimonio culturale, Archeologia, Architettura, Fabbricazione digitale e stampa 3D, Biologia e monitoraggio ambientale, Medicina e sistemi di eHealth, Moda del futuro, Sicurezza e difesa...

Programma:

- Modelli discreti per la rappresentazione dei dati in 3D (con un'attenzione particolare alle reti di triangoli) e tecniche di base di elaborazione 3D
- Fondamenti della Grafica al Computer (tecniche di rendering, illuminazione e texture)
- Librerie e strumenti disponibili



Figure 4: Esempi di utlizzo dei modelli 3D

### 1.3 Scientific Visualization

Tecniche informatiche per la generazione di rappresentazioni visuali interattive di dati spazio-temporali acquisiti o simulati (collegamento naturale con il mondo 3D+tempo in cui viviamo). Le illustrazioni a la comunicazione visuale della conoscenza è parte della nostra storia, anche oggi a causa dell'incremento della quantità dei dati da analizzare è necessario un approccio grafico per la visualizzazione. La Visualizzazione Scientifica si applica in diversi campi come l'Ingegneria, la medicina e la Scienza. La Visualizzazione Scientifica riguarda anche la definizione di algoritmi efficienti per la manipolazione interattiva ed esplorazione dei dati e delle loro caratteristiche.

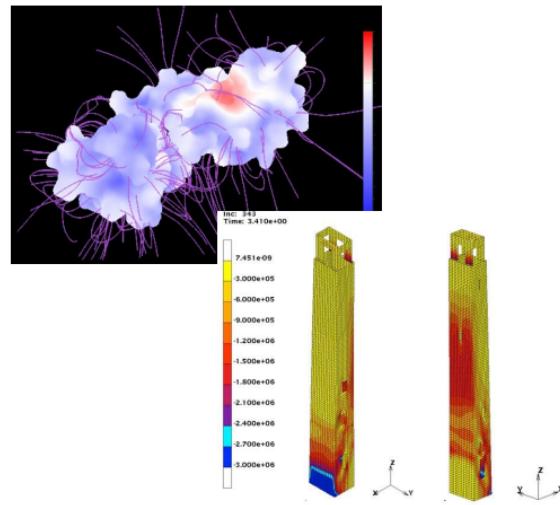


Figure 5: Esempi di Visualizzazione Scientifica

## 2 Information Visualization

La visualizzazione delle informazioni è un modo per comunicare significativamente i dati e aiutare le persone a dare un senso a grandi quantità di informazioni. Le visualizzazioni come artefatti cognitivi che consentono il trattamento parallelo delle informazioni. La visualizzazione ha tre scopi principali:

- **Explanatory:** Le visualizzazioni esplicative sono strumenti per presentare informazioni, comunicare dati e messaggi, spiegare qualcosa a qualcun altro.
- **Exploratory:** Le visualizzazioni esplorative sono strumenti per i lettori per analizzare ciò che viene loro presentato. Molto spesso, l'esito dell'analisi esplorativa non è solo la risposta alle domande originali, ma la generazione di nuove domande.
- **Confirmatory:** Nell'analisi confermativa, le visualizzazioni sono destinate a testare ipotesi.

Bisogna decidere cosa visualizzare

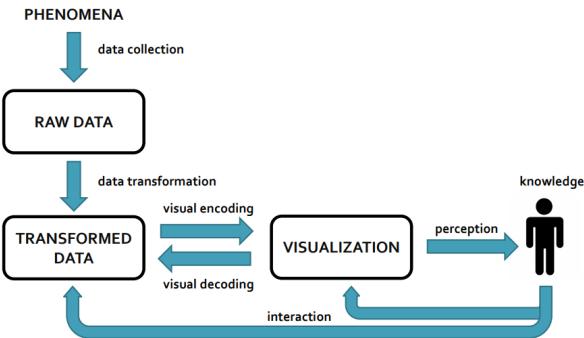


Figure 6: Visualization pipeline

### 2.1 Data

Informazioni fattuali come misurazioni o statistiche, utilizzate come base per il ragionamento, la discussione o il calcolo. I dati sono collezioni di *items* e *attributes* degli items. Gli *Items* sono gli oggetti/entità che vogliamo visualizzare. Gli *Attributi* sono le proprietà degli oggetti/entità. Una *Tabella* è una griglia di colonne e righe, dove le righe rappresentano gli item e le colonne gli attributi. Un *network* è una collezione di nodi che rappresentano gli item, connessi tramite dei link; entrambi nodi e link possono avere attributi.

### 2.2 Tipi di attributi

- *Quantitativi*: sono gli attributi dove i valori rappresentano quantità misurate, questi valori possono essere ordinati, ma può essere calcolata anche la distanza tra i valori.
- *Categorici*: sono gli attributi dove i valori descrivono le categorie, possono essere di tre tipi, **Nominali** se non hanno un ordine particolare, **Ordinali** se possono essere ordinati e **Binary** se hanno solo due stati.

## 2.3 Semantica degli attributi

- **Spaziali e temporali:** esempio la location, latitudine e longitudine o la data di assunzione come attributo temporale.
- **Sequnziali, ciclici divergenti:** esempio: i mesi dell'anno sono ciclici, la temperatura è un attributo divergente.
- **Gerarchici:** tipi di prodotto con sottocategorie, un esempio sono i vestiti.

Si deve selezionare la visualizzazione appropriata a seconda del tipo e dalla semantica dell'attributo. Esempio:

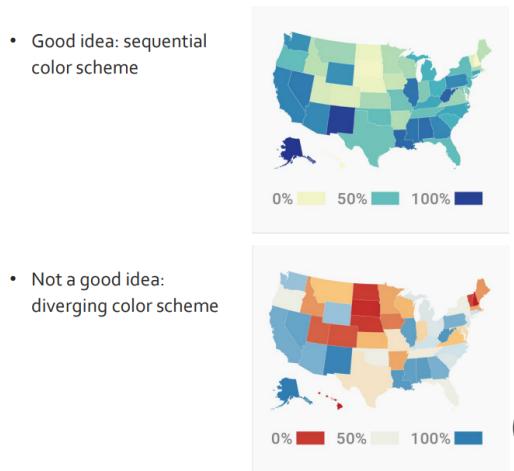


Figure 7: Differenza di Visualizzazione

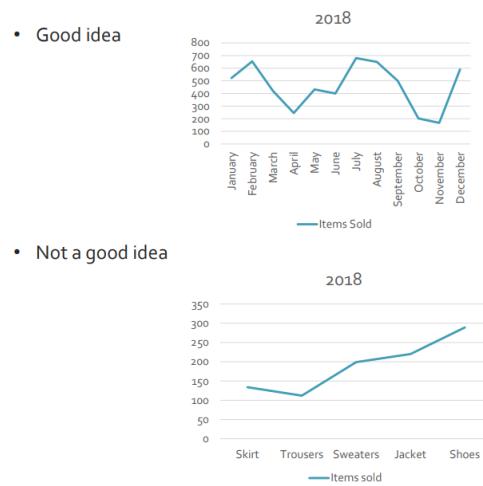


Figure 8: Differenza di Visualizzazione

## 2.4 Grafici Fondamentali (Bivariate data)

### 2.4.1 Bar Charts

Visualizza come una quantità misurata si distribuisce tra categorie. Ogni barra rappresenta una categoria e la lunghezza della barra è una quantità misurata in quella categoria. Dati bivariati: nominale/ordinale e quantitativo. Non confondere con gli istogrammi

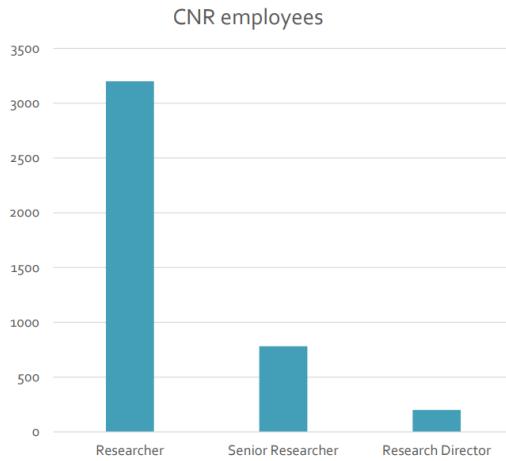


Figure 9: Bar Charts

#### 2.4.2 Histograms

Frequenza degli elementi di dati bivariati: una variabile indipendente quantizzata in intervalli (blocchi) e una variabile dipendente.

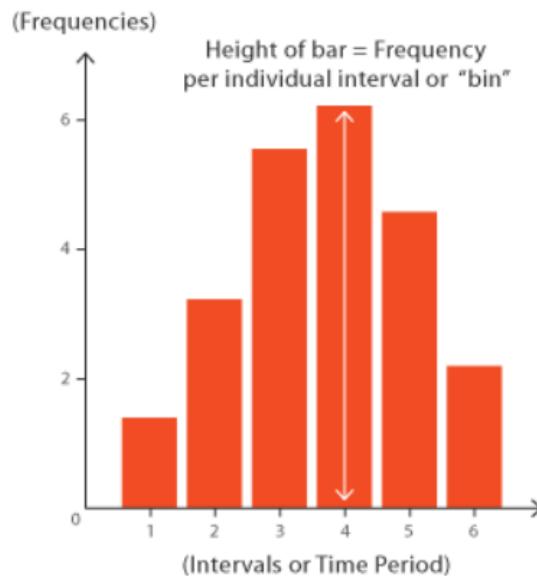
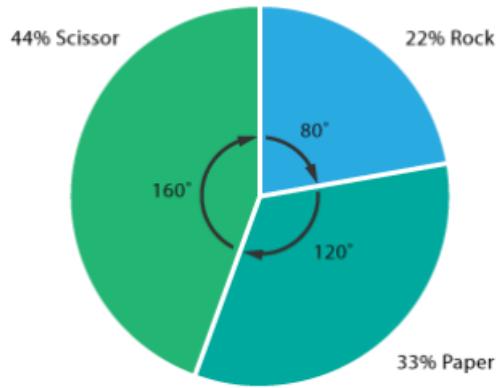


Figure 10: Histograms

#### 2.4.3 Pie Charts

Mostrare proporzioni e percentuali tra le categorie. Utile per dare un’idea rapida e confrontare una fetta rispetto al totale, non adatto per confronti accurati. Altri svantaggi: numero limitato di valori, occupazione dello spazio.



Data			
Rock	Paper	Scissor	TOTAL
2	3	4	9
To calculate percentages			
$2/9=22\%$	$3/9=33\%$	$4/9=44\%$	100%
Degrees for each "pie slice"			
$(2/9) \times 360 = 80^\circ$	$(3/9) \times 360 = 120^\circ$	$(4/9) \times 360 = 160^\circ$	360°

Figure 11: Pie Charts

#### 2.4.4 Donut Charts

Grafici a torta con l'area centrale tagliata. Maggiore enfasi sulla lunghezza dell'arco rispetto all'area. Più efficienti in termini di spazio.

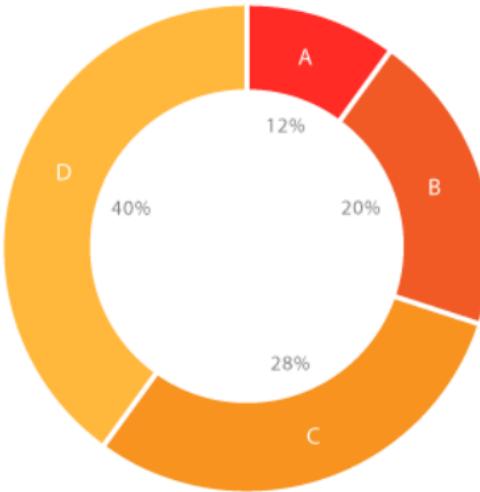


Figure 12: Donut Charts

## 2.4.5 Scatter plots

Relazione tra attributi: visualizzazione di come una quantità è correlata a un'altra (e analisi di cluster, valori anomali, ecc.). Dati bivariati: due attributi quantitativi indipendenti.

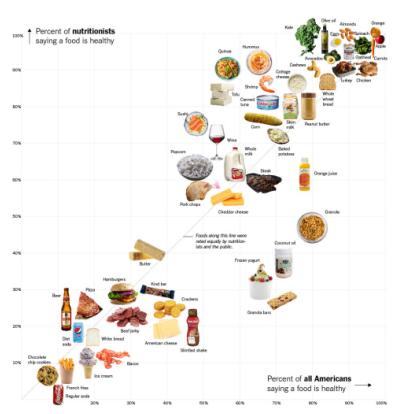


Figure 13: Scatter plots

## 2.4.6 Slope Charts

Alternativa ai Scatter plots: assi paralleli, ogni elemento è una linea che collega due quantità.

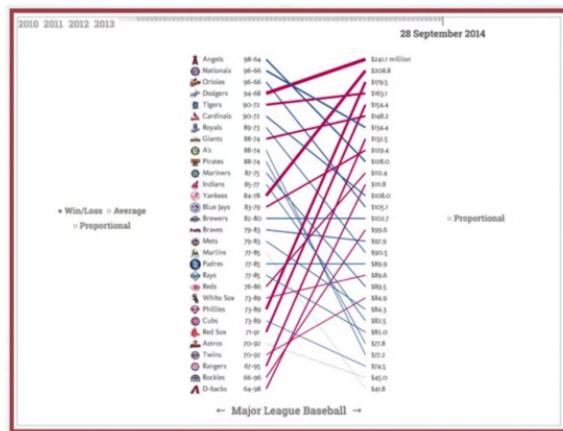


Figure 14: Slope Charts

#### 2.4.7 Line Charts

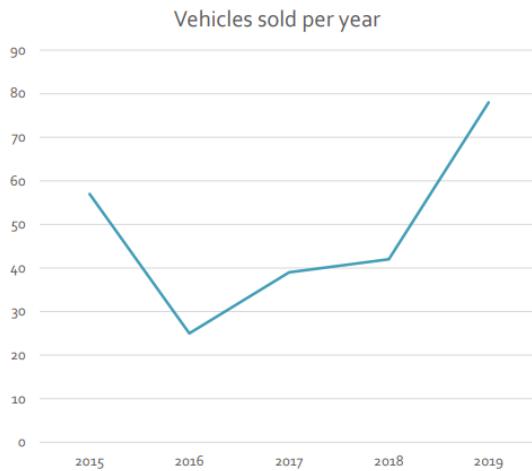


Figure 15: Line Charts

#### 2.4.8 Area Charts

Line Charts dove l'area sotto alla linea è riempita.

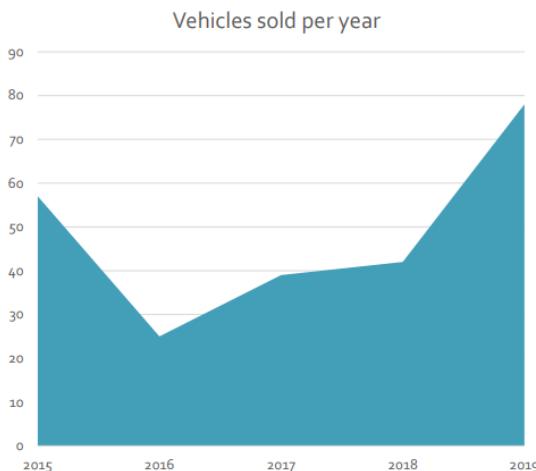


Figure 16: Area Charts

#### 2.4.9 Chropleth maps

Come si distribuisce una quantità nelle diverse aree/geografiche/regioni. Colori, sfumature, pattern sono utilizzati per rappresentare la quantità associata alle aree/regioni. Buona panoramica (ma non confronto accurato). Rischio: confondere l'area geografica con i valori dei dati (prestare attenzione alla normalizzazione).

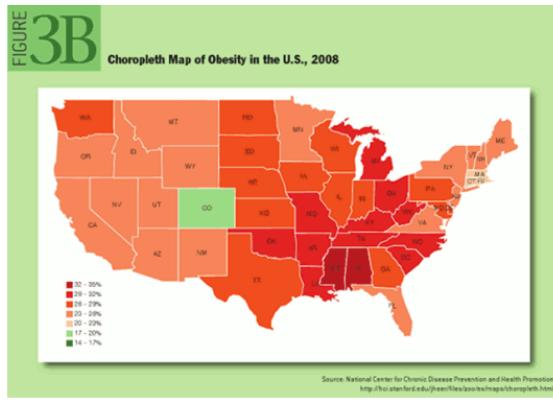


Figure 17: Choropleth maps

#### 2.4.10 Symbol maps

Come si distribuisce una quantità lungo due coordinate spaziali. Un simbolo (spesso un disco o un quadrato) è posizionato in un punto e dimensionato in modo che la sua area sia proporzionale alla quantità associata al punto.

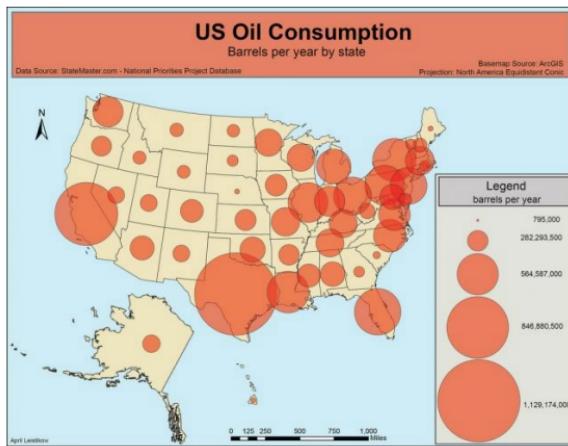


Figure 18: Symbol maps

### 2.5 Grafici Fondamentali (Attributi Multipli)

#### 2.5.1 Stacked Bar Charts

Rappresentazione di attributi aggiuntivi. Tante barre quanti sono le categorie del primo attributo, e altrettanti segmenti all'interno di ciascuna barra quanti sono i valori dell'altro attributo categorico. Utile per comunicare la proporzione

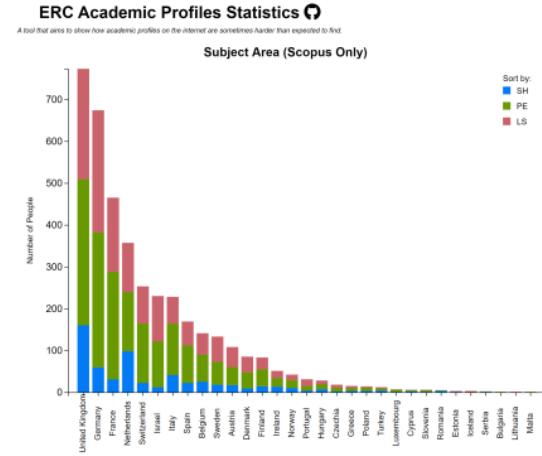


Figure 19: Stacked Bar Charts

### 2.5.2 Grouped Bar Charts

Rappresentazione di attributi aggiuntivi. Lo stesso grafico a barre viene ripetuto più volte per il numero di categorie nell’attributo aggiuntivo. Utile per confrontare i valori individuali

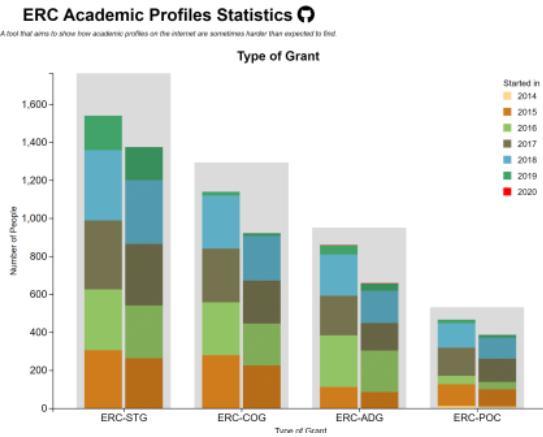
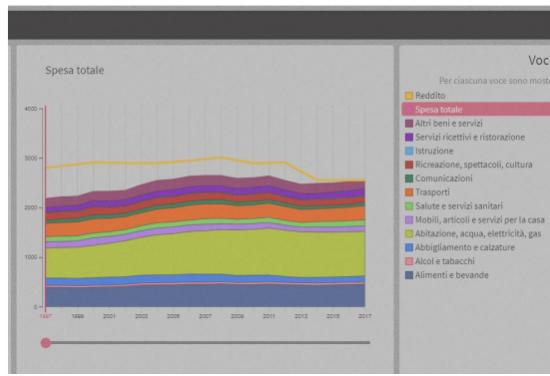


Figure 20: Grouped Bar Charts

### 2.5.3 Stacked Line Charts

I valori tra le categorie sono impilati uno sopra l’altro. Non è ideale per leggere e confrontare i valori individuali nel tempo: la forma delle linee è influenzata dalla forma delle linee sottostanti (interazione/filtraggio per compensare). Non è significativo per i numeri negativi e dati che non dovrebbero essere sommati (ad esempio, temperatura).



#### 2.5.4 Line Charts Series

Raccolta di linee nello stesso grafico. Facile confrontare i valori, ma le curve sovrapposte possono rendere il grafico difficile da leggere

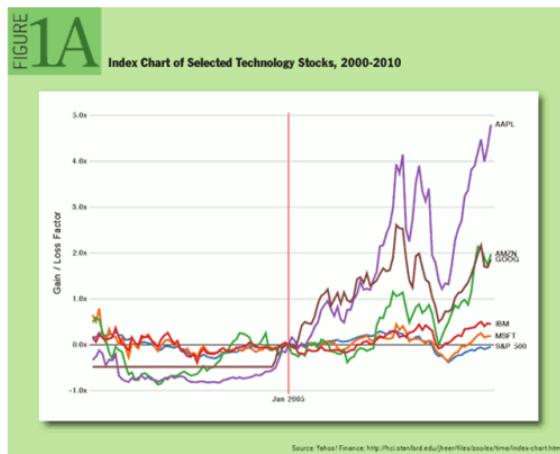


Figure 21: Line Charts Series

#### 2.5.5 Bubble Charts

Variante degli scatterplot, adattata per rappresentare attributi aggiuntivi. Le aree dei cerchi rappresentano un attributo aggiuntivo (avviso: area, non raggio). I colori possono distinguere le categorie o rappresentare attributi aggiuntivi. Capacità limitata di dimensioni dei dati (interazione per compensare).

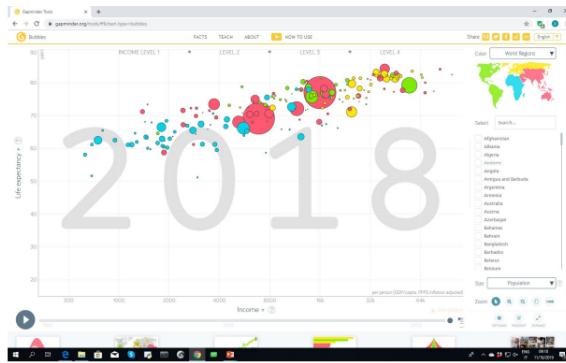


Figure 22: Bubble Charts

### 2.5.6 Small multiples

Valido per tutti i tipi di grafici.

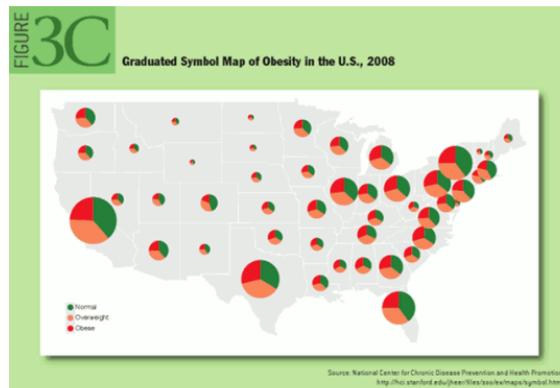


Figure 23: Small multiples

### 2.5.7 Symbol Maps

Questo tipo di mappa utilizza simboli, come icone o forme, per visualizzare informazioni diverse relative a diverse caratteristiche o attributi su una mappa geografica.

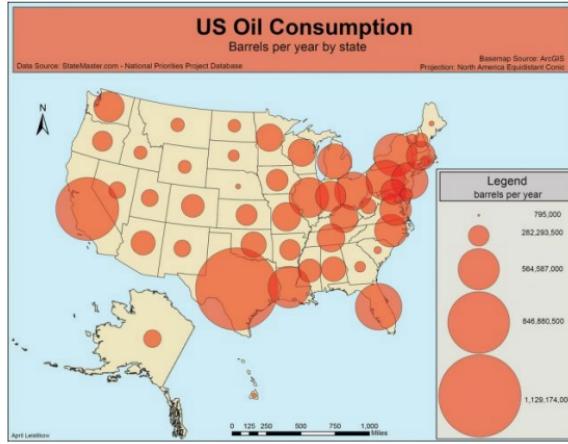


Figure 24: Symbol Maps

## 2.6 Grafici Fondamentali (Attributi Gerarchici)

### 2.6.1 Sunburst Charts

Per mostrare gerarchie, ogni anello rappresenta un livello nella gerarchia, muovendosi verso l'esterno.

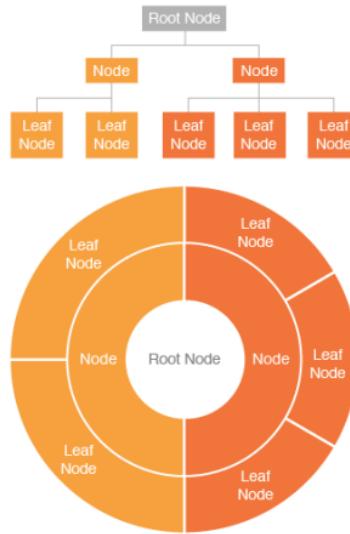


Figure 25: Sunburst Charts

### 2.6.2 Treempas

Un modo alternativo per visualizzare le gerarchie, utilizzando la dimensione dell'area. Ogni rettangolo rappresenta una categoria, con rettangoli sottocategorie nidificati. Quando una quantità è assegnata a una categoria, la sua dimensione dell'area viene visualizzata in proporzione a quella quantità e alle altre quantità all'interno della stessa categoria padre in una relazione parte-tutto. Diverse strategie di impaginazione (ad esempio, algoritmo squarified). Compatto e efficiente nello spazio, ottimo per avere una

panoramica delle strutture, perfetto per confrontare proporzioni tramite la dimensione dell'area, ma meno efficace nel mostrare i livelli nella gerarchia rispetto, ad esempio, ai grafici sunburst charts.

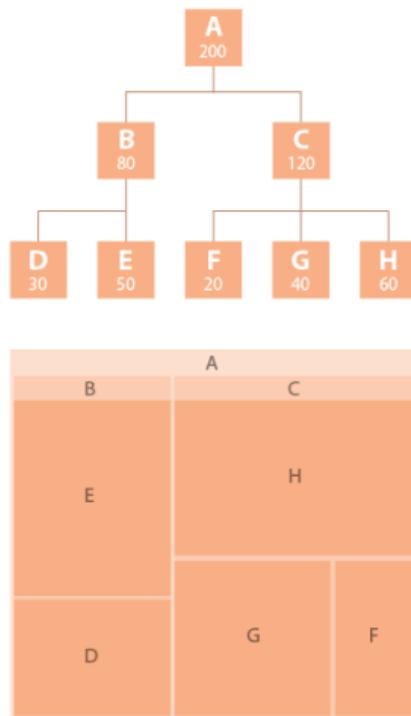


Figure 26: Treempas

# 3 Information Visualization II

## 3.1 Visual Encoding

Le visualizzazioni sono composte da **marks** (segnali) e **channels** (canali). I marks sono gli oggetti grafici che rappresentano gli elementi di dati (ad esempio, punti, linee, barre). I channels sono le proprietà grafiche che rappresentano gli attributi dei dati (ad esempio, colore, posizione, forma, dimensione).

La codifica visiva significa passare dai dati alle rappresentazioni visive. La codifica richiede non solo la scelta di un grafico appropriato per i dati in questione (tipo e semantica), ma anche la selezione degli elementi grafici individuali e delle loro proprietà.

## 3.2 Graphical Elements

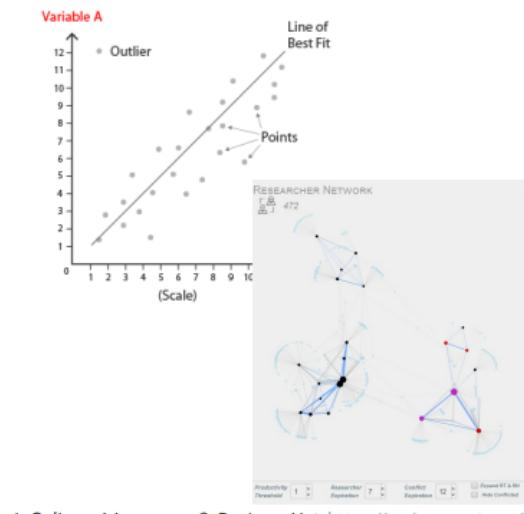


Figure 27: Esempio di Marks

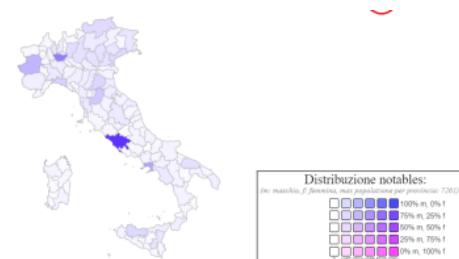


Figure 28: Esempio di Channels

Le componenti contestuali sono elementi che rendono più semplice interpretare le visualizzazioni. Un esempio sono: le labesls, annotazioni, legende, griglie, assi cartesiani.

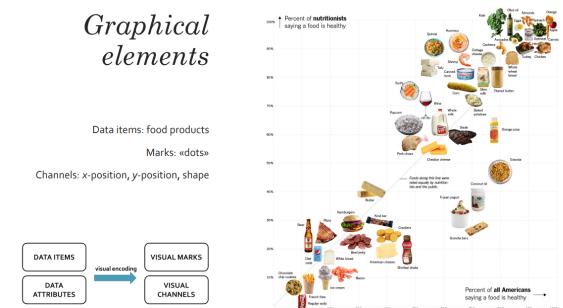


Figure 29: Esempio completo di Channels e Marks di un grafico

### 3.3 Visual decoding

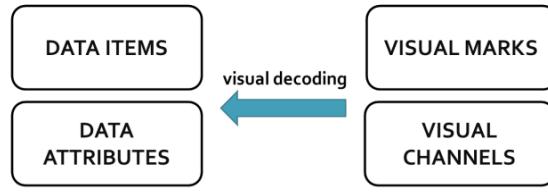


Figure 30: Visual Decoding

Per **Visual Decoding** si intende destrutturare una rappresentazione visiva nei suoi principali elementi e identificare:

- gli elementi grafici, quali sono i segnali visivi? Quali sono i canali visivi?
- le regole di mappatura (ossia, le informazioni che i singoli elementi grafici rappresentano) quali elementi di dati rappresentano i segnali? Quali attributi rappresentano i canali?

È utile per valutare e ridisegnare le visualizzazioni.

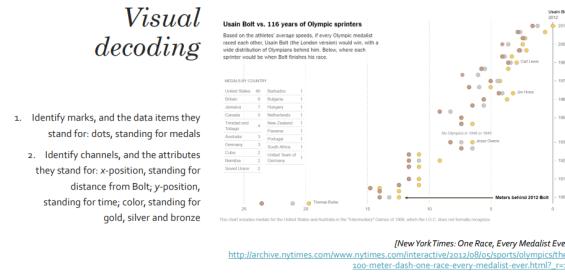


Figure 31: Esempio completo di Decoding

#### 3.3.1 Quality Evaluation

La valutazione di una visualizzazione può basarsi su due principi guida principali: **Expressiveness** ed **Effectiveness**. **Expressiveness**: la Visual representation dovrebbe rappresentare tutte e solo le relazioni che esistono nei dati. Le informazioni rilevanti dovrebbero essere prioritarie e quindi codificate con i canali più efficaci/accurati.

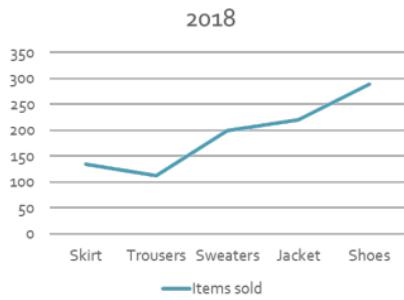


Figure 32: Dati non ordinati che sembrano ordinati (nell'esempio seguente, stiamo mostrando informazioni su un trend che non è nei dati: la forma della linea non ha significato).

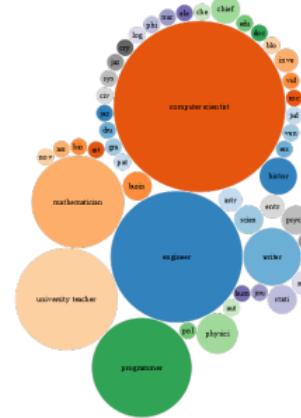


Figure 33: utilizzo dei colori quando non restituiscono alcuna informazione

### 3.3.2 Percentuale di Accuracy

Quanto sono efficaci i canali nel trasmettere diversi tipi di attributi? Uno dei (possibili) riassunti per attributi quantitativi.

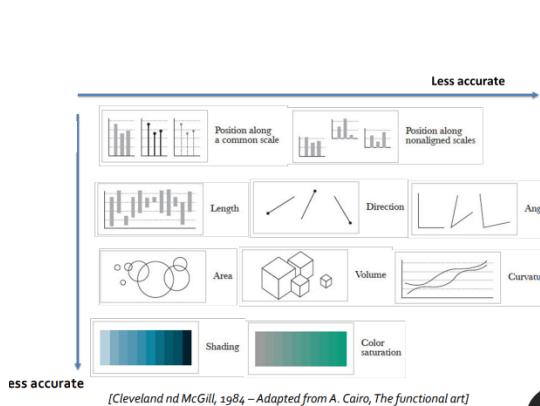


Figure 34: Riassunto dell'accuracy di ogni grafico.

QUANTITATIVE	ORDINAL	CATEGORICAL
Position	Position	Position
Length	Density	Hue
Angle	Saturation	Texture
Slope	Hue	Connection
Area	Texture	Containment
Volume	Connection	Density
Density	Containment	Saturation
Saturation	Length	Shape
Hue	Angle	Length
	Slope	Angle
	Area	Slope
	Volume	Area
		Volume

[Data Visualization Course by John C. Hart, for Coursera, 2015]

Figure 35: Riassunto dell'accuracy di ogni grafico

## 3.4 Altri tipi di grafici

### 3.4.1 WordClouds

Rappresentazione di quanto frequentemente le parole appaiono in un determinato corpo di testo attraverso la dimensione della parola. Variazioni nell'arrangiamento e nel colore. Principalmente utilizzato per motivi estetici.



Figure 36: WordClouds

### 3.4.2 Calligrams

Testi disposti in modo tale da formare un'immagine tematicamente correlata. L'immagine creata dalle parole illustra il testo esprimendo visivamente qualcosa associato (o in contrasto) a ciò che il testo dice.



Figure 37: Calligrams

### 3.4.3 Flow Maps

Raffigurare il movimento delle entità posizionando linee tracciate sopra delle mappe (spazio e tempo) Lo spessore, il colore, ecc. possono codificare informazioni aggiuntive In questa mappa: dimensioni delle truppe, distanza percorsa, temperatura, latitudine e longitudine, direzione del viaggio, tempo.

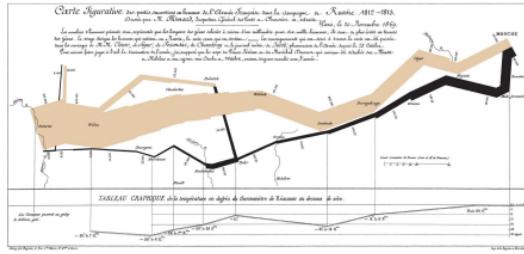


Figure 38: Flow Maps

### 3.4.4 Chernoff faces

Ragionamento: siamo molto bravi nel riconoscere i volti. Introdotto da Herman Chernoff nel 1973, le variabili sono mappate su tratti del viso (larghezza/curvatura della bocca, dimensione verticale del viso, dimensione/inclinazione/separazione degli occhi, dimensione delle sopracciglia, posizione verticale delle sopracciglia...)

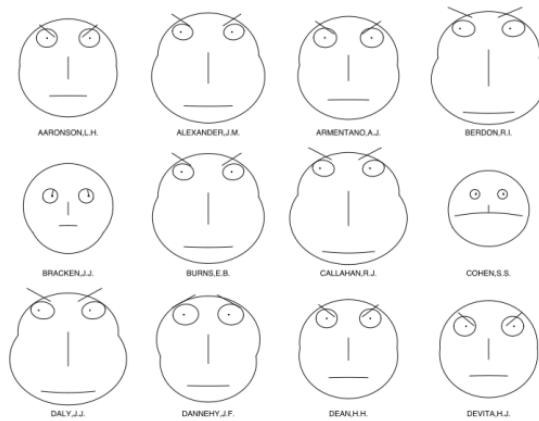


Figure 39: Chernoff faces

### 3.4.5 Multidimensional Icons

Spence e Parr (1991) proposero di codificare le proprietà di un oggetto in una semplice rappresentazione tramite icone. Hanno applicato questo approccio per verificare le offerte di permanenza.

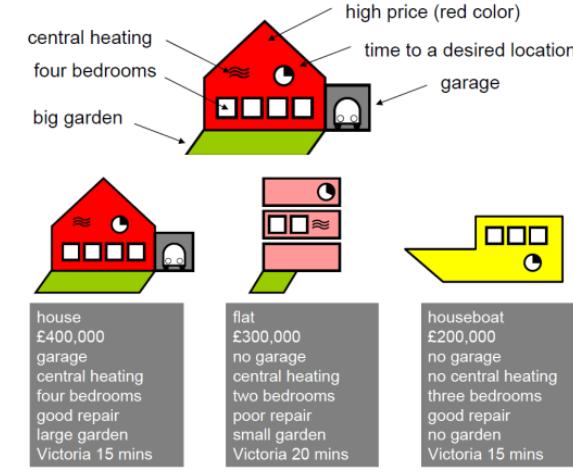


Figure 40: Multidimensional Icons

### 3.4.6 Petal as a gliph

L'idea di Moritz Stefaner per visualizzare un indice di vita consiste nel mappare diverse variabili (relative alle condizioni di vita materiale e alla qualità della vita) in petali di diverse dimensioni, per confrontare il benessere tra paesi.

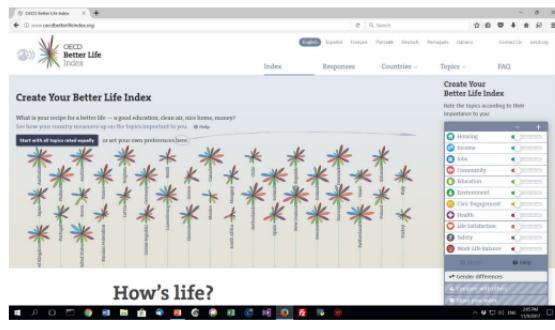


Figure 41: Petal as a gliph

## 4 Applied Perception

Abbiamo visto che la visualizzazione delle informazioni consiste nel trasformare i dati in una rappresentazione visuale in modo che un essere umano possa estrarre informazioni utili da essi. L'**Effectiveness** di una rappresentazione visuale non è arbitraria: ma dipende fortemente su come funziona il cervello. Remind: l'efficacia si riferisce alla capacità di una rappresentazione visiva, come un grafico, una mappa o un diagramma, di comunicare in modo chiaro e comprensibile le informazioni desiderate. Capire come funziona la percezione può aiutare a prendere decisioni informate riguardo ai design delle visualizzazioni. L'occhio umano svolge un ruolo fondamentale nella percezione visiva, essendo il principale organo sensoriale che ci consente di interpretare il mondo circostante. L'**acuità visiva** è una misura della capacità dell'occhio di distinguere dettagli fini e di percepire chiaramente gli oggetti. Viene spesso misurata in termini di precisione nella lettura di lettere o simboli su una tavola ottotipica posta a una certa distanza. La nostra percezione è sensibile al contrasto di pattern, alla frequenza e all'orientamento. Inoltre, il colore influenza sulla funzione del sistema di contrasto spaziale (CSF). Il visual Cortex è una parte del cervello che gestisce l'elaborazione delle informazioni visive provenienti dagli occhi.

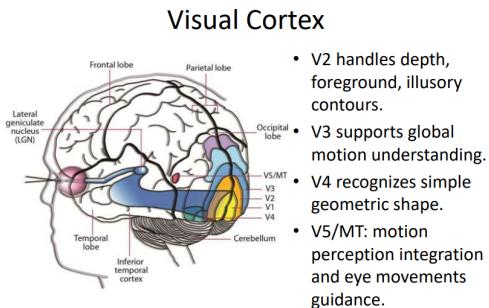


Figure 42: Divisione del cervello

### 4.1 Illusions

Il campo recettivo di una cellula è l'area visiva sulla quale una cellula risponde alla luce. Le cellule gangliari della retina sono organizzate con campi recettivi circolari. Quando vengono stimolate al centro, vengono eccitate; quando vengono stimolate al di fuori del centro, vengono inibite.

#### 4.1.1 Mach Banding

Le "Mach bands" (bande di Mach) sono un fenomeno ottico che si verifica quando si osservano transizioni tra regioni di luce e ombra su una superficie, creando percezioni di contrasto accentuato lungo i confini.

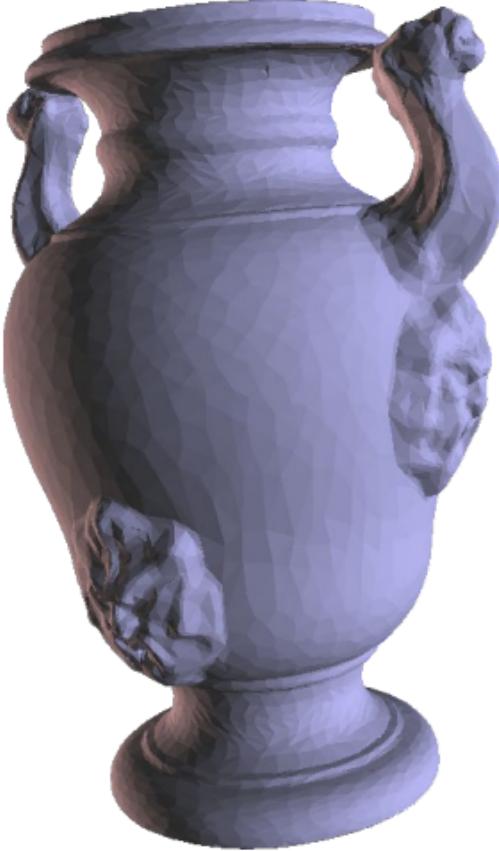


Figure 43: Mach Banding

#### 4.1.2 Hermann Grid illusions

La griglia di Hermann è un'illusione ottica che mette in evidenza l'interazione tra le cellule retiniche e il modo in cui il cervello elabora le informazioni visive. Questa illusione è composta da una griglia di linee grigie disposte a intervalli regolari, con dei punti neri posizionati strategicamente agli incroci delle linee. Si ritiene che l'illusione sia causata dalla percezione del contrasto e dall'inibizione laterale nel sistema visivo. Le cellule retiniche inviano segnali al cervello che possono essere interpretati in modo errato, facendo percepire punti ombrosi o grigi nei punti di intersezione delle linee.

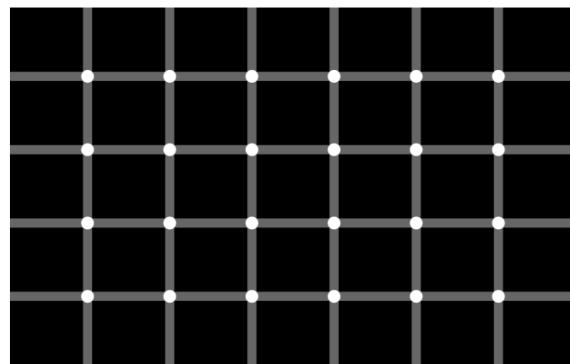


Figure 44: Divisione del cervello

#### 4.1.3 Chevreul Illusion

L'illusione di Chevreul, chiamata anche illusione dei contrasti simultanei, è un fenomeno ottico che coinvolge la percezione del colore. Questa illusione è dovuta alla sensibilità del nostro sistema visivo ai contrasti e alla relativa interpretazione dei colori circostanti. Nell'illusione di Chevreul, se viene disegnata una serie di linee parallele in tonalità di grigio simile su uno sfondo bianco, la percezione visiva può far sembrare che i toni di grigio varino lungo le linee. Questo accade a causa dell'interazione tra i colori circostanti e la nostra percezione del contrasto. Se due tonalità di grigio simili sono separate da una riga più chiara, il grigio sembrerà più scuro del suo vero colore. Al contrario, se sono separate da una riga più scura, sembrerà più chiaro. Questo effetto può far sembrare che i toni di grigio cambino o che vi sia un gradiente lungo le linee parallele, quando in realtà la differenza di colore non esiste realmente.



Figure 45: Chevreul illusion

#### 4.1.4 Greyscale Maps

Questi effetti visivi possono portare a grandi errori quando si leggono informazioni quantitative visualizzate utilizzando una mappa in scala di grigi. Utilizzare mappe in scala di grigi per rappresentare pochi valori (!)

#### 4.1.5 The Cornsweet Effect

Le illusioni di Cornsweet sono un tipo di illusione ottica che evidenzia la percezione umana del contrasto e dei confini tra diverse superfici o tonalità. Questo tipo di illusione si basa sul modo in cui il cervello elabora i cambiamenti graduali di luminosità o colore. Nell'illusione di Cornsweet, due aree contigue hanno un confine graduale tra loro e una transizione graduale di luminosità o colore. Nonostante la transizione sia graduale e non ci sia un cambiamento netto, la percezione visiva può far sembrare che ci sia un confine più netto o un cambiamento repentino.

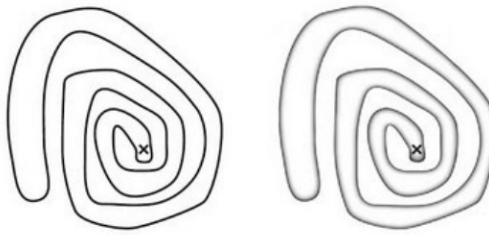


Figure 46: Cornsweet Effect

L'effetto Cornsweet può essere utilizzato per evidenziare regioni delimitate. Parlando dell'utilizzo delle scale di grigi, in sostanza ci sono delle regole da seguire: Non utilizzare per mappe o per confrontare molti valori. Utilizzare per evidenziare:

- Regioni delimitate
- Elementi importanti (riducendo il contrasto luminoso degli elementi non importanti)
- Regolare la luminanza dello sfondo per ottenere una migliore leggibilità.

## 4.2 Eye Movements

**Movimenti oculari:**

- **Movimenti saccadici:** movimenti balistici degli occhi che cambiano il punto di fissazione. Possono essere volontari o scatenati da stimoli.
- **Movimenti di inseguimento lento:** movimenti lenti degli occhi per mantenere uno stimolo in movimento sulla fovea.
- **Movimenti di vergenza:** allineano la fovea di ciascun occhio a un bersaglio in base alla sua distanza.
- **Movimenti vestibolo-oculari:** stabilizzano gli occhi compensando i movimenti della testa.

### 4.2.1 Preattentive Processes

I processi preattentivi si riferiscono a quelle operazioni cognitive che avvengono automaticamente e istantaneamente nell'elaborazione delle informazioni sensoriali prima che una persona focalizzi la sua attenzione in modo consapevole su di esse. Questi processi avvengono in modo rapido e automatico, senza richiedere uno sforzo cosciente. Sono responsabili della capacità del cervello di elaborare estrarre informazioni visive, uditiva o tattili in modo veloce ed efficiente. Questi processi sono fondamentali nell'organizzazione delle informazioni sensoriali e possono avvenire a livelli inconsci della percezione umana. Caratteristiche visive elaborate preattivamente:

- Orientamento ; Curvatura ; Forma ; Dimensione ; Colore ; Luce/Ombra ; Contenimento ; Concavità/Convessità ;

- Alcune di queste non sono simmetriche. Caratteristiche visive non elaborate preattivamente: Giunzione ; Parallelismo

Alcuni processi preattentivi non sono simmetrici: Aggiungere segni è più efficiente che rimuovere segni. Aumentare la nitidezza è più efficiente che diminuire la nitidezza. Un oggetto grande circondato da oggetti piccoli è più efficiente di un oggetto piccolo circondato da oggetti grandi.

#### 4.2.2 Gestalt Laws

Le leggi della Gestalt sono principi fondamentali della psicologia della percezione che descrivono come il cervello umano organizza e interpreta le informazioni sensoriali per creare una percezione significativa e coerente del mondo che ci circonda.

1. **Legge della Prossimità:** Gli oggetti vicini tra loro tendono ad essere visti come parte di un insieme o di un gruppo.
2. **Legge della Similarità:** Gli elementi simili per forma, colore, dimensione o altre caratteristiche tendono ad essere raggruppati insieme nella percezione.
3. **Legge della Connessione:** Gli oggetti connessi sono percepiti come correlati. Collegare diversi oggetti con una linea è un modo efficace per esprimere che esiste una relazione tra di essi.
4. **Legge della Continuità:** Gli oggetti posti lungo una linea o una curva tendono ad essere visti come parte di un modello o di un flusso continuo.
5. **Legge della Simmetria:** Gli oggetti disposti simmetricamente sono percepiti come formare un insieme visivo anziché essere considerati entità separate. La simmetria è meglio percepita per gli assi orizzontali e verticali.
6. **Legge della Chiusura:** Le persone tendono a percepire le figure incomplete come forme complete riempiendo le parti mancanti.
7. **Legge della Simplicità o della Buona Forma:** La percezione tende a organizzare gli stimoli in modo da formare figure semplici e coerenti piuttosto che configurazioni disordinate o complesse.
8. **Legge della Destinzione Figure-Ground:** Gli oggetti possono essere percepiti come figure distinte rispetto al loro sfondo circostante.
9. **Legge del Movimento Comune:** Gli oggetti che si muovono nella stessa direzione o seguendo lo stesso modello di movimento possono essere percepiti come parte di un unico oggetto o gruppo.

## 4.3 Illusions

### 4.3.1 Muller-Lyer

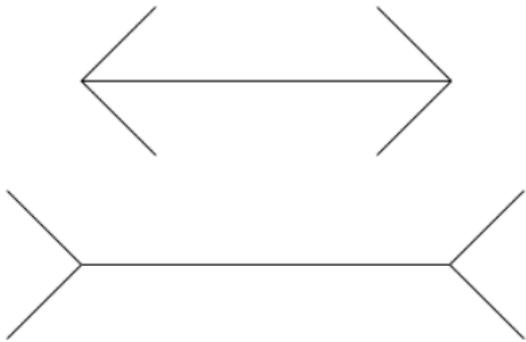


Figure 47: Muller-Lyer Illusions

Queste due linee hanno lunghezze uguali ma vengono percepite di lunghezze diverse.

- Due spiegazioni:
  1. Spiegazione prospettica
  2. Spiegazione del baricentro

### 4.3.2 Wundt

Queste illusioni coinvolgono principalmente la percezione delle dimensioni, delle proporzioni e delle distanze tra gli oggetti, portando a percezioni distorte o erronee della realtà visiva.

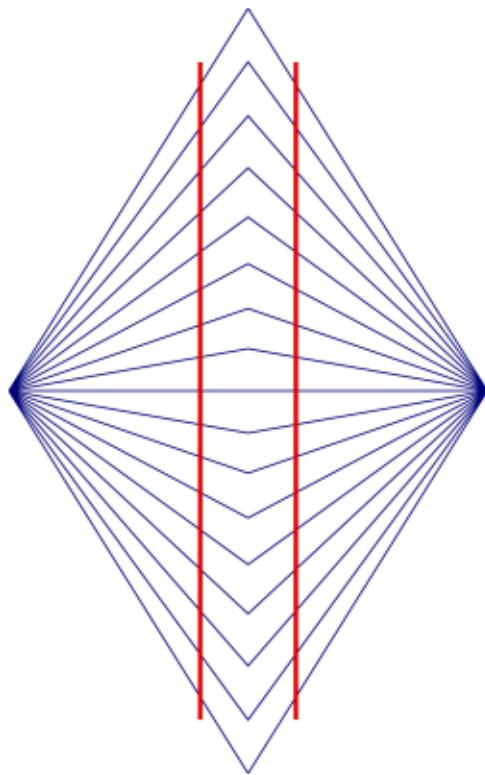


Figure 48: Muller-Lyer Illusions

#### 4.3.3 Hering

Un'altra illusione simile (effetto invertito dell'illusione di Wundt).

- Possibili spiegazioni:
  1. Inibizione laterale
  2. Effetto prospettico
  3. Ritardi temporali nell'elaborazione visiva

#### 4.3.4 Horizontal–Vertical Illusion

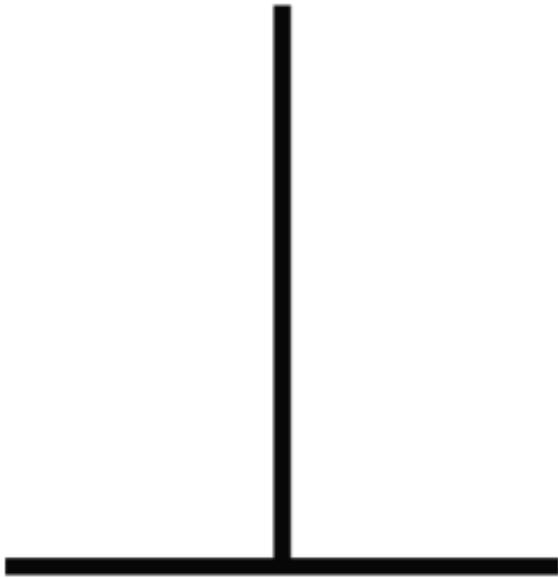


Figure 49: Horizontal–Vertical Illusion

Un'altra semplice illusione scoperta da Wundt. La linea verticale viene percepita come lunga al 30% in più rispetto alla linea orizzontale. Sono state osservate differenze (piccole) tra le culture. Questo vale anche per le linee che si intersecano.

#### 4.3.5 Comparing Area

Confrontare le aree è difficile (ricordare le aree dei cerchi appena menzionate).

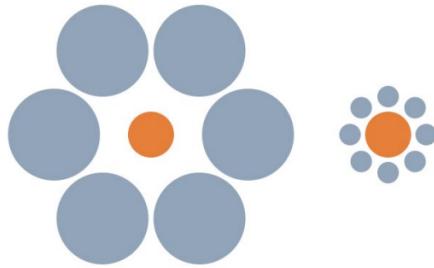


Figure 50: Comparing Area

- Quando confrontiamo le aree, le proporzioni sono sottovalutate (peggio per i volumi).
- Flannery (1970) propose di compensare la percezione applicando un fattore di scala percettiva.

- Tufte, nel suo famoso libro "The Visual Display of Quantitative Information" (2001), si oppose a qualsiasi cosa tranne che l'uso di una scala assoluta, ovvero esclude la compensazione per le mancanze percettive umane.

La scalatura percettiva potrebbe essere insufficiente. Le cose sono più complesse da un punto di vista percettivo

#### 4.3.6 Weber's Law

Ernst Heinrich Weber (1795 1878) ha condotto studi sulla percezione degli stimoli fisici da parte dei sensi umani (visione, udito, gusto, tatto e olfatto). Legge di Weber:

$$\frac{\Delta S}{S} = K$$

Legge di Weber:

- La percezione dipende dallo stimolo iniziale.
- I rapporti sono più importanti dei valori assoluti.

# 5 Applied Perception II

## 5.1 Color Vision

La *Color Vision* può essere considerata come superflua nella vita moderna, ma il colore comunque è estremamente utile nella data visualization in diversi casi:

- Mostrare i patterns
- Labeling
- Mettere in evidenza



Figure 51: Effetti della color vision

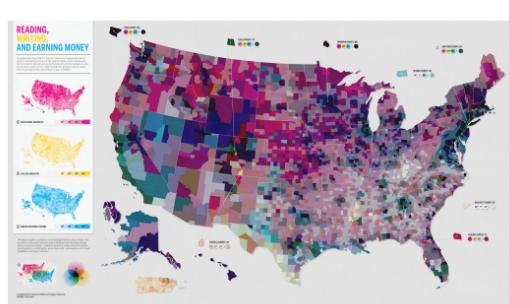


Figure 52: Effetti della color vision

Pensare al colore come un attributo di un oggetto invece che ad una caratteristica primaria(C.Ware).

## 5.2 Trichromacy and Opponent process theories

Abbiamo, sulla retina, tre distinti recettori per i colori, **icones** (coni) che sono attivi ai livelli normali di luce e sono fotorecettori sensibili alla luce intensa e al colore, mentre i **rods** (bastoncelli) non hanno influenze sulla percezione, sono fotorecettori sensibili alla luce scarsa e responsabili della visione notturna e periferica.

### 5.2.1 Trichromacy theory

I *cones* sono sensibili alle diverse onde di luce. Quindi, assorbono luce attorno lo spettro del colore blue, verde e rosso. La teoria dice che percepiamo il colore tramite un sistema a tre canali. Tutti gli spazi dei colori, anche se progettati per scopi diversi, sono tridimensionali.

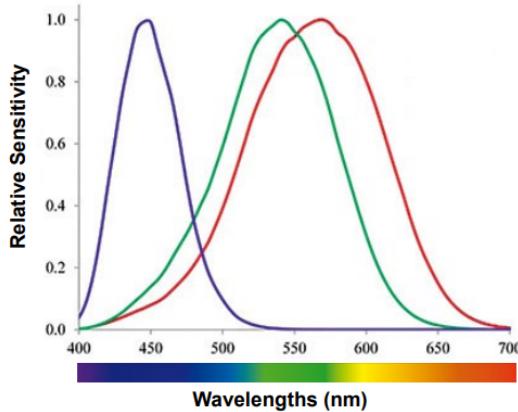


Figure 53: Trichromacy theory

### 5.2.2 colour measurement and specification

Dato che solo tre diversi recettori sono coinvolti nella colori vision, è possibile fare il match dei colori usando un mix di tre colori, chiamati primari. Dato uno standard di colori primari, si può usare una trasformazione per creare lo stesso colore in output in device diversi

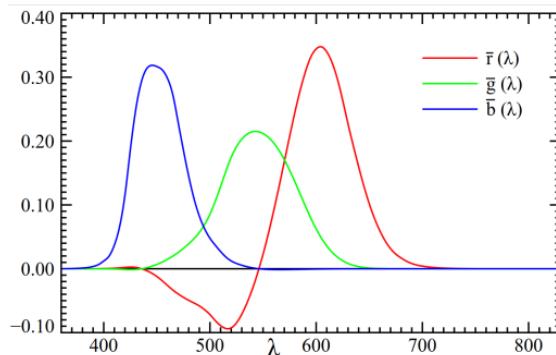


Figure 54: RGB color Matching Function

### 5.2.3 CIE XYZ Color Space

Lo spazio colore CIE XYZ è uno spazio colore standardizzato sviluppato dalla Commission Internationale de l'Eclairage (CIE). È un modello matematico che rappresenta tutti i colori visibili agli esseri umani attraverso tre valori numerici: X, Y e Z. Questi valori sono definiti in base alle risposte dei tre tipi di coni presenti nell'occhio umano (sensibili alle lunghezze d'onda dei colori rosso, verde e blu). Il valore Y rappresenta la luminosità, mentre X e Z descrivono la posizione orizzontale e verticale del colore nello spazio XYZ. Per un valore fisso di Y, il piano XZ contiene tutte le possibili cromaticità a quella luminanza.

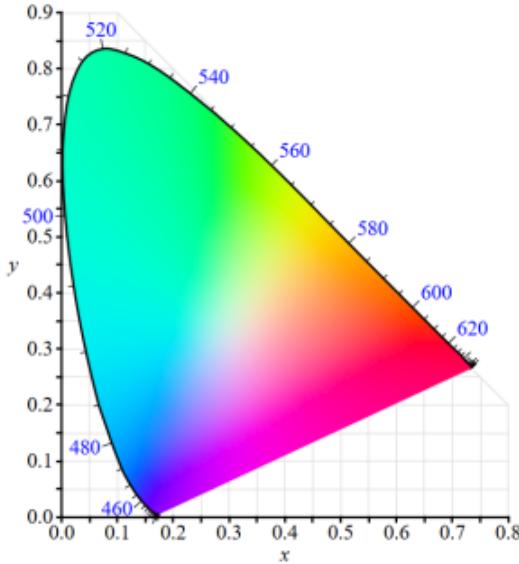


Figure 55: Chromaticity diagram

Il diagramma di cromaticità è uno strumento visivo utilizzato per rappresentare i colori visibili senza tener conto della luminosità. Questo diagramma rappresenta le proprietà del colore in termini di tonalità e saturazione, ignorando la luminanza.

#### 5.2.4 Opponent process theory

Alla fine del XIX secolo, il psicologo tedesco Hering propose la teoria (successivamente supportata da prove sperimentali) secondo cui i coni della retina combinano il loro stimolo formando tre coppie di colori che competono tra loro per formare quello finale. Queste coppie, chiamate coppie antagoniste, sono: nero-bianco, giallo-verde e giallo-blu

**Evidenze a supporto della teoria:**

- Nomenclatura
- Tonalità uniche
- Neurofisiologia

**Proprietà dei canali di colore antagonisti:**

- Risoluzione spaziale
- Percezione della forma
- Contrasto cromatico

La teoria della tricromia e la teoria dei colori antagonisti operano a livelli differenti.

La teoria della tricromia spiega ciò che accade a livello dei fotorecettori. La teoria dei colori antagonisti spiega ciò che accade a livello neurale.

## 5.3 Color Space

### 5.3.1 RGB Color Space

Lo spazio dei colori **RGB** (Red, Green, Blue) è un modello di colore utilizzato comunemente nel contesto digitale e dell'informatica per rappresentare i colori. Questo spazio dei colori si basa sulla combinazione di tre colori primari: rosso (Red), verde (Green) e blu (Blue). Nel modello RGB, ogni colore può essere rappresentato come una combinazione di intensità di questi tre colori primari. Ogni colore è rappresentato da un punto in uno spazio tridimensionale dove i tre assi corrispondono a rosso, verde e blu. La combinazione di diverse intensità di rosso, verde e blu consente di ottenere una vasta gamma di colori. Ad esempio, il nero si ottiene quando tutte e tre le componenti (R, G, B) sono al minimo, mentre il bianco si ottiene quando tutte e tre sono al massimo. I colori primari (rosso, verde e blu) sono usati come base per creare tutti gli altri colori nel modello RGB.

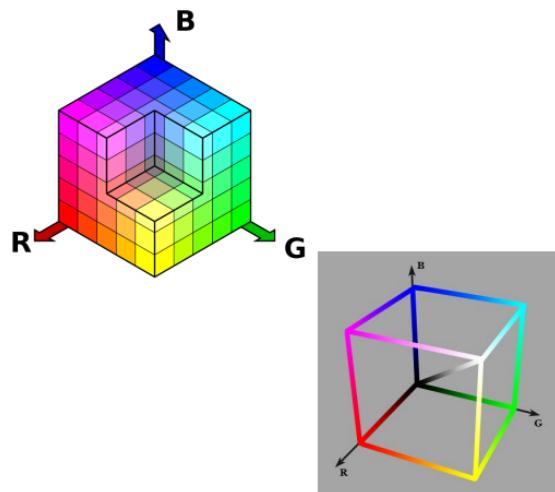


Figure 56: RGB color

### 5.3.2 HSV/HSL color space

**HSV** (Hue, Saturation, Value) e **HSL** (Hue, Saturation, Lightness) sono due modelli di spazio dei colori correlati che rappresentano i colori in termini di tre componenti principali: tonalità (Hue), saturazione (Saturation) e valore o luminosità (Value per HSV, Lightness per HSL). La specifica del colore risulta essere più intuitiva con HSV/HSL rispetto a RGB. Tuttavia, questi spazi di colore non sono uniformi dal punto di vista percettivo, il che implica che le distanze calcolate all'interno dello spazio colore non corrispondono alle distanze percettive.

### 5.3.3 CIE Lab/Lch color space

Lo spazio colore CIE Lab/Lch è un sistema di colore sviluppato dalla Commissione Internazionale dell'Illuminazione (CIE). Questo spazio colore è progettato per rappresentare i colori in modo più uniforme e vicino alla percezione umana rispetto ad altri spazi colore come RGB, HSV o HSL.

### 5.3.4 Color differences

Avere uno space color in cui le distanze percettive uguali corrispondono a distanze uguali è utile per specificare tolleranze del colore, codici colore, pseudocolore (utilizzando sequenze di colori per rappresentare valori dei dati, possibilmente con passaggi percettivamente uguali). Anche se gli space color uniformi forniscono solo un'approssimazione iniziale approssimativa di come saranno percepite le differenze di colore, un fattore influente importante è la dimensione (siamo molto più sensibili alle differenze tra grandi campi). Suggerimento: utilizzare colori saturi quando si codifica piccoli simboli o linee sottili e colori meno saturi per aree grandi.

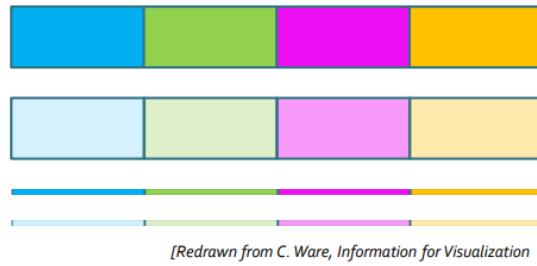


Figure 57: Color differences

## 5.4 Color and visualization

### 5.4.1 Luminance and visualization

I canali cromatici rosso-verde e giallo-blu sono in grado ciascuno di trasportare solo circa un terzo della quantità di dettagli trasportati dal canale in bianco e nero (Mullen, 1985).

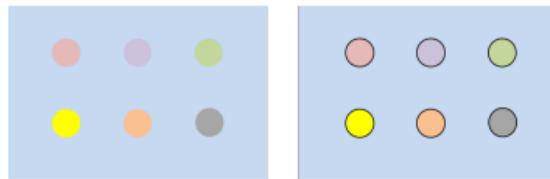


Figure 58: Luminance and visualization

Le differenze puramente cromatiche non sono sufficienti per visualizzare dettagli fini. Assicurare un adeguato contrasto di luminanza con lo sfondo (anche se vengono utilizzati colori con differente cromaticità). Un confine di contrasto può migliorare la leggibilità dei simboli colorati.

### 5.4.2 Saturation and visualization

Utilizzare colori saturi per codificare piccoli simboli/dettagli fini e colori meno saturi per codificare aree grandi.

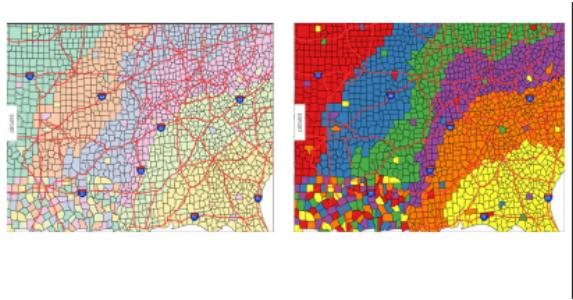


Figure 59: Saturation and Visualization

#### 5.4.3 Color for Labeling

Post e Greene (1986) hanno condotto un esperimento sulla denominazione dei colori (sono stati mostrati 210 colori diversi su uno sfondo nero in una stanza oscurata). Solo otto colori più il bianco vengono denominati in modo coerente. Anche se non è generalmente applicabile, ciò suggerisce che solo pochi colori possono essere utilizzati come etichette di categoria.

#### 5.4.4 Color and semantics

Fai attenzione alle convenzioni dei colori e alle associazioni semantiche (rosso per caldo/cattivo/pericolo, blu per freddo, verde per vita/vai, ecc.), poiché le convenzioni non sono universali. L'associazione semantica con il grigio è quella di appartenenza a una categoria non specificata (utile per evidenziare) Evita colori problematici per le persone daltoniche. Utilizza una scala di colori basata sullo spettro solo quando il suo utilizzo è profondamente radicato nella cultura degli utenti. Per rivelare dettagli fini, utilizza sequenze di pseudocolori che variano nella luminosità, non solo nella cromaticità.

#### 5.4.5 Color Ramps

Evita le scale di colore problematiche per le persone con daltonismo. Utilizza una scala di colori basata sullo spettro quando il suo utilizzo è profondamente radicato nella cultura degli utenti. Per rivelare dettagli precisi, utilizza sequenze di pseudocolori che variano nella luminosità, non solo nella cromaticità.

## 6 Multidimensional Data

### 6.1 Multi-Set Bar Charts

I "Multi-set Bar Chart" sono grafici a barre che mostrano più insiemi di dati su uno stesso grafico, consentendo di confrontare più categorie o gruppi di dati attraverso barre separate. In un grafico a barre tradizionale, ogni barra rappresenta un singolo insieme di dati, ad esempio una singola categoria o una serie temporale. Tuttavia, nel caso dei multi-set bar charts, diversi insiemi di dati vengono rappresentati con barre raggruppate insieme per ogni categoria o punto temporale. Le barre per ogni categoria sono divise in sezioni separate, ognuna rappresentante un insieme di dati diverso. Questo tipo di grafico è particolarmente utile quando si desidera confrontare le relazioni tra diverse categorie attraverso più serie di dati. Ad esempio, immagina un grafico a barre in cui ogni barra rappresenta le vendite totali mensili di diversi prodotti in un negozio. Le barre sono divise in segmenti colorati, ognuno dei quali rappresenta le vendite mensili di un prodotto specifico. Ciò consente di visualizzare facilmente le vendite totali di ogni mese e di confrontare le performance dei vari prodotti all'interno dello stesso periodo. Questi grafici possono essere utilizzati in molteplici contesti, come analisi di mercato, confronti tra diversi gruppi di dati, analisi delle prestazioni in settori come vendite, finanza, ricerca scientifica e altro ancora.

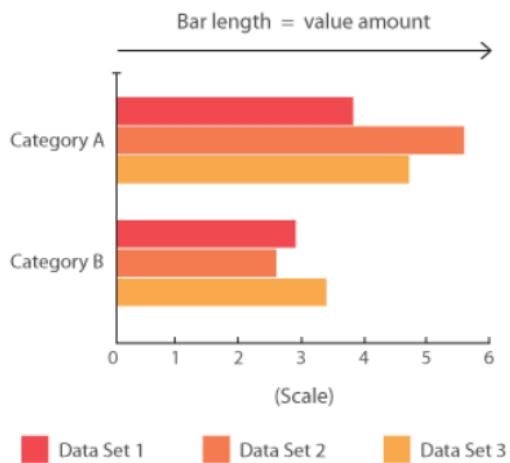


Figure 60: Multi-Set Bar charts

### 6.2 Multiple Bars

Distribuzioni di ciascuna variabile tra le diverse categorie/punti dati (ogni variabile ha la propria visualizzazione).

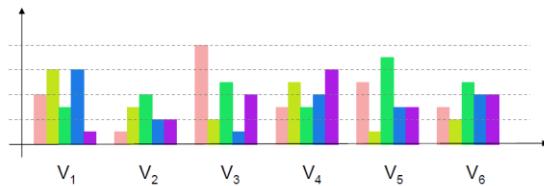


Figure 61: Multiple Bars

### 6.3 Stacked Bar Charts

grafici a barre sovrapposte" o "grafici a barre impilate". Questi grafici mostrano più serie di dati sovrapposte o impilate verticalmente all'interno di ciascuna categoria o punto dati. Ogni barra rappresenta l'insieme completo dei dati per quella categoria e le diverse sezioni della barra corrispondono alle diverse componenti o sotto-categorie di quella categoria. Simple Stacked Bar Charts: - Utile se la visualizzazione dei valori assoluti (e la loro somma) ha significato. Percentage Bar Charts: Migliori per mostrare le differenze relative tra le quantità nei diversi gruppi.

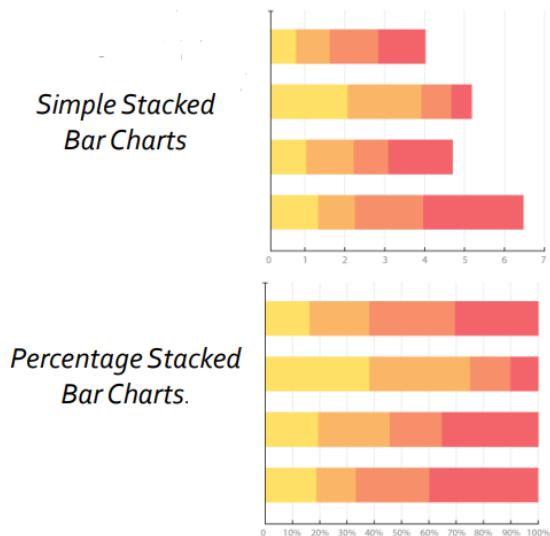


Figure 62: Two types of Stacked Bar

### 6.4 SpinePlots

Gli spineplots, anche conosciuti come back-to-back plots, sono grafici utilizzati per visualizzare la distribuzione congiunta di due variabili categoriche. Questi grafici sono particolarmente utili per esaminare le relazioni tra due variabili qualitative. La caratteristica principale degli spineplots è quella di mostrare la distribuzione incrociata delle categorie delle due variabili, disposte in modo da rivelare le associazioni tra di esse.

## 6.5 Mosaic Plots

Un Mosaic Plot è un tipo di grafico statistico utilizzato per visualizzare la struttura relativa tra due o più variabili categoriche. Questo tipo di grafico è particolarmente utile per esaminare le relazioni tra variabili categoriche e mostrare la distribuzione proporzionale delle categorie in modo visivo. **Vantaggi:**

- Massimo utilizzo dello spazio disponibile
- Buona panoramica delle proporzioni tra i dati
- Buona panoramica della dipendenza delle variabili

**Svantaggi:**

- È difficile estendere il grafico a molte variabili

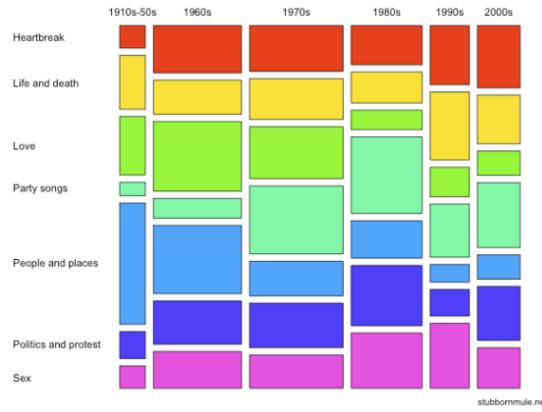


Figure 63: Mosaic Plots

## 6.6 Tree Maps

Un modo alternativo per visualizzare una struttura dati ad albero. Efficiente nello spazio (!) Il modo in cui i rettangoli sono divisi e ordinati in sottorettangoli dipende dall'algoritmo di suddivisione utilizzato.



Figure 64: Mosaic Plots

## 6.7 Parallel Coordinates

**Parallel Coordinates** è una tecnica di visualizzazione che consente di rappresentare graficamente relazioni complesse tra molteplici variabili. È utilizzato principalmente per esplorare e visualizzare dataset multivariati

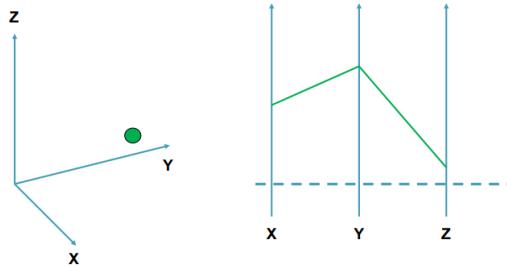


Figure 65: Mosaic Plots

## 6.8 Star Plot

Conosciuto con molti nomi: grafico radar, grafico a ragnatela, grafico a rete, ecc. Analogamente alle coordinate parallele, ma gli assi sono posizionati in coordinate polari (equi-angolari). La posizione del primo asse non fornisce informazioni rilevanti. Facile per confrontare le proprietà di una classe di oggetti o di una categoria. Non è facile comprendere il compromesso tra diverse variabili. Non adatto a molte variabili o a molti dati.

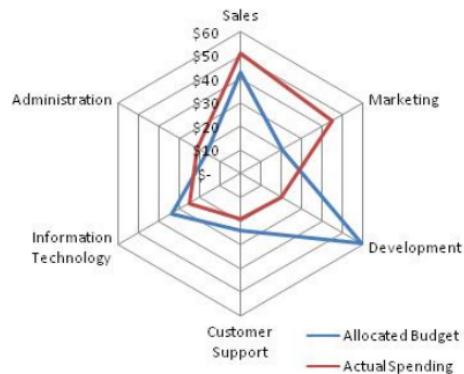


Figure 66: Star Plot

## 7 Dimensionality Reduction

Ragionamento: I dati N-dimensionali vengono proiettati in 2 o 3 dimensioni per una migliore visualizzazione/comprendimento. Strategia ampiamente utilizzata. In generale, si tratta di un mapping e non di una trasformazione geometrica. Diversi mappings presentano diverse proprietà.

### 7.1 Principal Component Analysis (PCA)

Una tecnica classica per la riduzione della dimensione è la **Principal Component Analysis (PCA)**. Si tratta di una tecnica lineare e non parametrica, l'idea principale è trovare una base formata da delle direzioni che massimizzano la varianza dei dati. L'idea è di rappresentare i dati in nuova base che meglio *esprime* i nostri dati. La nuova base è una combinazione lineare della base originale.

$$\mathbf{P}\mathbf{X} = \mathbf{Y}$$

Figure 67: Cambio di Base

#### 7.1.1 Signal-to-noise Ratio (SNR)

- Given a signal with noise:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

- It can be expressed as:

$$SNR = \frac{\sigma_{signal}^2}{\sigma_{noise}^2}$$

Figure 68: Basis

## 7.2 Covariance Matrix & Risolvere PCA

$\text{Cov}(x) = \dots$  Matrice quadrata simmetrica, gli elementi sulla diagonale corrispondono alla varianza di una particolare variabile. Come selezionare la  $P$  migliore per il cambio di base? Quella che minimizza la ridondanza e che massimizza la varianza. L'obiettivo è diagonalizzare la matrice della covarianza di  $Y$ . Valori alti sulla diagonale significano che la "dinamica" della singola variabile è stata massimizzata. valori bassi dei valori che non si trovano sulla diagonale significano che la ridondanza tra le variabili è minimizzata.

**Theorem:** una matrice simmetrica  $A$  può essere diagonalizzata da una matrice formata dai suoi autovettori come  $A = EDE'$ , dove le colonne di  $E$  sono gli autovettori di  $A$ . Quindi si organizzano i dati come una matrice  $mxn$ , si sottrae la media corrispondente ad ogni riga si calcolano gli autovettori e gli autovalori della matrice  $XX'$  e infine si organizzano per formare la matrice  $P$  L'idea è trovare la  $k$ -esima componente principale, proiettare i dati in queste direzioni e usare questi dati invece di quelli originali. Questi dati sono la migliore approssimazione rispetto alla somma dei quadrati delle differenze.

### 7.2.1 limit of PCA

Uno dei limiti è che la funzione non è parametrica -> è sia un punto forte che un punto debole Non funziona per i dati che non seguono una distribuzione gaussiana, può essere estesa per prendere in considerazione anche le trasformazioni non lineari.

### 7.2.2 PCA and MDS

## 7.3 Locally Linear Embedding

(LLE) **LLe** cerca di scoprire strutture non lineari in grandi dimensioni sfruttando le approssimazioni lineari locali.

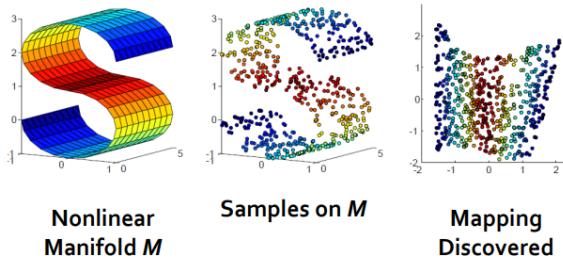


Figure 69: Locally Linear Embedding

Assumendo che ci sono dati sufficienti, ci aspettiamo che ogni punto e i suoi vicini possano essere approssimati da una patch lineare locale. La patch è rappresentata dalla somma pesata dei punti locali. Prendi un insieme di dati vicino ad un punto(ball-radius) e risolvi per  $W_{ij}$ . .... Trova il vettore  $Y_i$  che minimizza la funzione di costo (embedding).

- 1. Calcola i vicini di ogni data point  $X_i$ .
- 2. Calcola i pesi che meglio ricostruiscono  $X_i$ .

- 3. Calcola i vettori che minimizzano la funzione di costo.

## 7.4 ISOMAP

L'idea principale è quella di preservare la distanza geodesica tra due punti La distanza geodesica la distanza più breve tra due punti in uno spazio curvo.

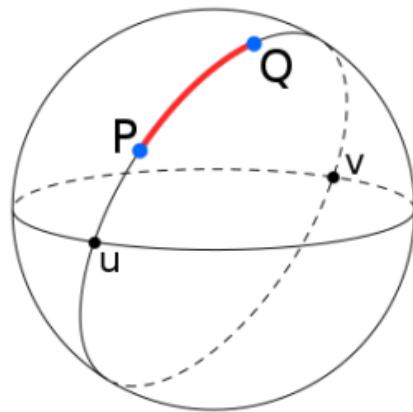


Figure 70: Geodesic distance

- 1. Costruire il grafo dei vicini.
- 2. Calcola il path più breve.
- 3. Costruisci il d dimensional embedding(?)

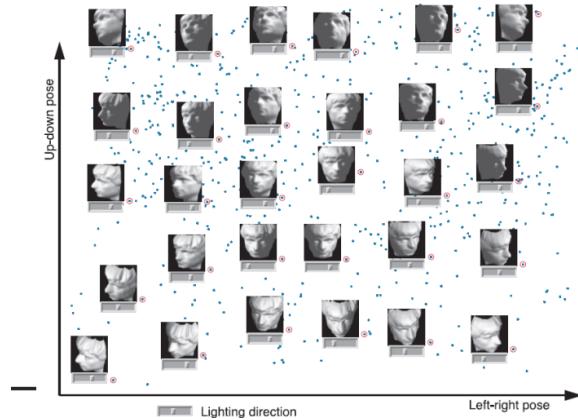


Figure 71: Geodesic distance

## 7.5 Autoencoders

Il Machine Learning ormai è presente in ogni ambito dell'informatica, un tipo speciale di neural network è chiamato **Autoencoder**. Un autoencoder può essere usato per la dimensionality reduction

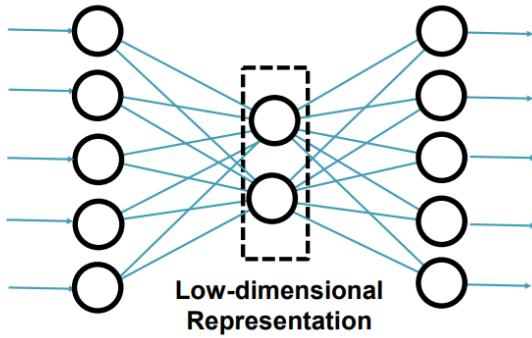


Figure 72: Geodesic distance

## 7.6 SNE, Stochastic Neighbor Embedding

Molte tecniche per la riduzione della dimensione non sono utili per restituire sia la struttura locale che quella globale dei dati in una singola mappa. Le Similarità tra high and low dimensional data points sono modellate con le condizioni di probabilità ...

# 8 3D Data Representation

Modellazione manuale utilizzando software dedicati, simulazioni Tecniche di acquisizione attive o passive (scansione laser, fotogrammetria). I progressi nelle tecniche di acquisizione hanno finalmente superato il 'bottleneck della modellazione': la creazione di modelli digitali 3D è diventata più semplice e meno costosa.

## 8.1 Gemoetry Processing

La disciplina riguardante modelli matematici per rappresentare dati geometrici e gli algoritmi per analizzarli e manipolarli.

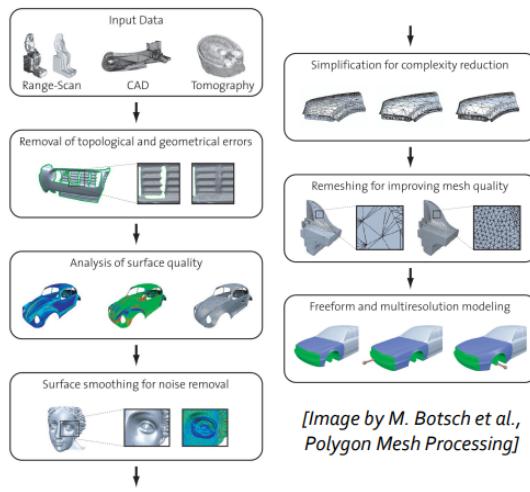


Figure 73: Gemoetry Processing

### 8.1.1 Surface in Computer Graphics

Definizione intuitiva in CG: Una regione bidimensionale in uno spazio tridimensionale, che ha lunghezza e larghezza ma nessuno spessore. Se chiusa, rappresenta una forma tridimensionale (il confine di un volume).

Surfepicie implicita:

$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

$$S(x, y, z) = f(x, y, z) = 0$$

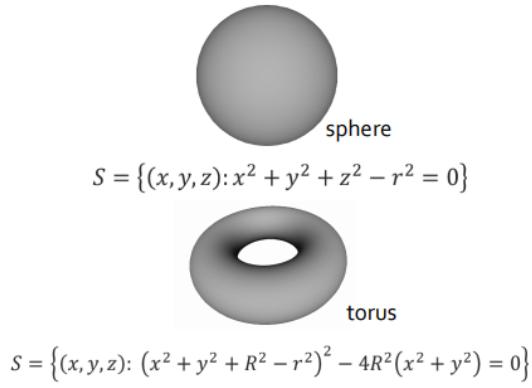


Figure 74: Sphere and torus

### Parametri Surface

Rappresentare un oggetto analiticamente? Ok, ma cosa succede con gli oggetti del mondo reale? Gli oggetti del mondo reale sono difficili da definire in modo analitico. Cosa succede agli oggetti del mondo reale? Esistono molte rappresentazioni discrete alternative, a seconda di: tradizione, origine dei dati, comodità per il compito e spazio di archiviazione (dimensioni). **Idea:** forme complesse possono essere rappresentate come collezioni di elementi di base. Prendi una serie di elementi semplici e uniscili insieme (in modo strutturato).

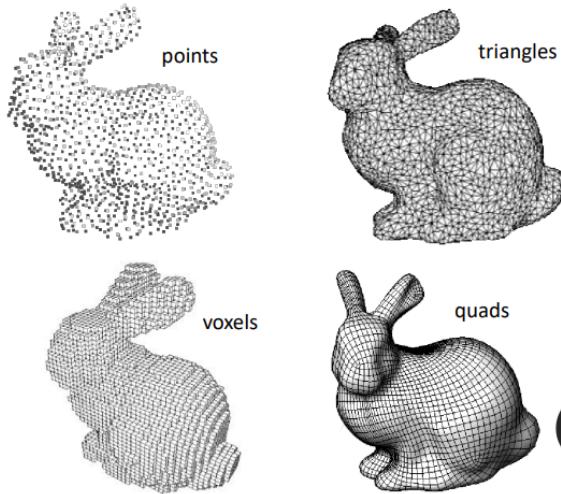


Figure 75: Rappresentazioni in modo strutturato

#### 8.1.2 Voxels

Considera le superfici come il confine degli oggetti volumetrici solidi. Definisci una funzione indicatrice che mappa il dominio tridimensionale in un dominio binario discreto (1 significa solido e 0 significa vuoto). Memorizza la funzione in una griglia regolare di voxel. Anche La **Signed Distance Field** (SDF) viene memorizzato in una griglia di voxel. Offre più informazioni rispetto a una semplice mappa di occupazione. Maggior spazio di archiviazione

- Pro: Struttura regolare
- Contro: Potenzialmente storage massivi

Le griglie di voxel possono memorizzare: funzioni indicatrici, campi di distanza (segmenti), campi generici (ad esempio, densità, vettori). Maggiore è la risoluzione, maggiore è la dimensione.

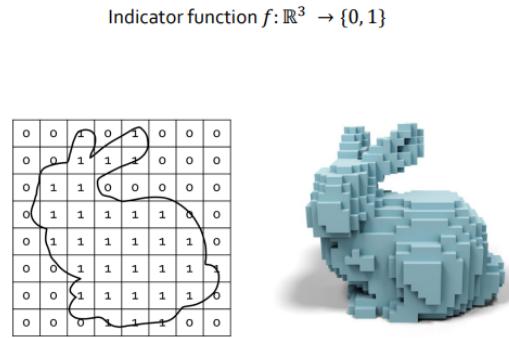


Figure 76: Rappresentazioni con voxels

### 8.1.3 Point Clouds

Collezione non ordinata di sample di punti,

- **l'origine dei dati:** 3d scans, Lidar, Structure-from-Motion
- **Tasks:** Renderig veloce di dataset di grandi dimensioni, processi iterativi e storage di raw data.

Uno dei pro è che i punti sono compatti e facili da gestire e tra i vari contro abbiamo che i punti sono irregolari, inconsistenti, non rappresentano davvero superfici.

point 1	x	y	z
point 2	x	y	z
point 3	x	y	z
point 4	x	y	z
			⋮
point N	x	y	z

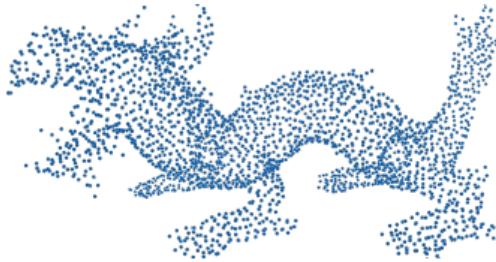


Figure 77: Rappresentazioni con voxels

#### 8.1.4 Polygonal meshes

I poligoni sono *building blocks*.

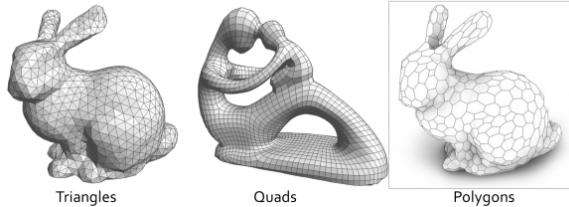


Figure 78: Rappresentazioni con i Poligoni

I triangoli come building blocks: Una primitiva molto concisa, con molti vantaggi (pianarità e convessità, potenza di approssimazione...). Rappresentazione di superfici de facto standard per il trattamento della geometria classica.

- **Origine:** modelli CAD, artisti 3D, dati di scansione 3D elaborati, ecc.
- **Compiti:** Rendering 3D (film, videogiochi), applicazioni interattive (GPU), modellazione 3D, elaborazione delle superfici 3D, ...
- **Pro:** Rappresentazione adeguata delle superfici 3D, discretizzazione adattiva, modo più efficiente di memorizzare superfici 3D generiche.

Ma comunque questo metodo è ancora Irregolare, non ordinato, inconsistente, non orientato.

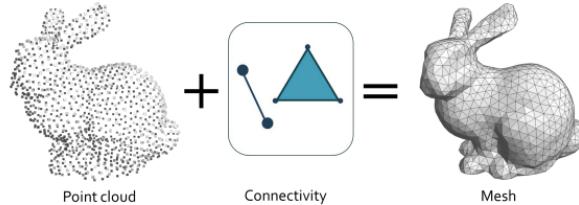


Figure 79: Traingel Meshes

Vertex Matrix			Face Matrix			
vertex 1	x	y	triangle 1	1	2	3
vertex 2	x	y	triangle 2	3	2	4
vertex 3	x	y	triangle 3	4	2	5
vertex 4	x	y	triangle 4	7	6	5
:			:			
vertex N	x	y	triangle M	10	87	30

Figure 80: Vertex Matrix and Face Matrix

### 8.1.5 Simplicies

Gli elementi fondamentali sono chiamati **Simplices**. Un  $k$ -simplex  $\Delta_k$  in  $\mathbb{R}^n$ ,  $0 \leq k \leq n$ , è l'involturo convesso di  $k+1$  punti affinamente indipendenti  $A_0, A_1, \dots, A_k$  chiamati vertici. In  $\mathbb{R}^3$  ci sono 4 possibili **Simplices**: Punto (0-simplex); Segmento di retta (1-simplex); Triangolo (2-simplex); Tetraedro (3-simplex). Un simplesso può essere orientato assegnando un ordine ai suoi vertici. Due orientamenti che differiscono per una permutazione pari determinano una stessa orientazione (negativa o positiva).

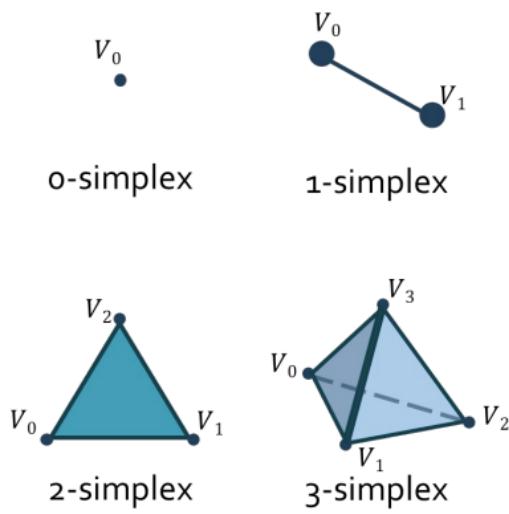


Figure 81: Vari tipi di Simplices

### 8.1.6 Simplicies complexes

Una faccia (corretta) di un  $k$ -simplex  $\Delta_k$  è un simplex il cui insieme di vertici è un sottinsieme non vuoto dell'insieme di vertici di  $\Delta_k$ . Ad esempio, per un 1-simplex/segmento: i suoi vertici/estremi; per un 2-simplex/triangolo: sia i suoi spigoli/segmenti che i vertici. Un complesso simpliciale finito  $\mathcal{K}$  è una collezione finita di simplessi che si incontrano solo lungo una faccia comune, oltre ai loro facce di qualsiasi dimensione.

- $\mathcal{K}$  is a simplicial complex if
  - if  $A, B \in \mathcal{K}$ , then  $A \cap B$  is either empty or is a face of both  $A$  and  $B$
  - if  $A \in \mathcal{K}$  and  $B$  is a face of  $A$ , then  $B \in \mathcal{K}$

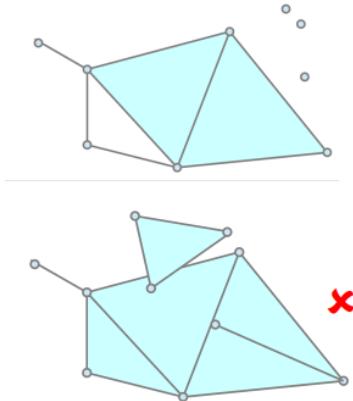


Figure 82: Simplicial Complex

La dimensione del complesso simpliciale è la massima dimensione dei suoi simplessi. Un complesso simpliciale  $k$ -dimensionale è omogeneo se è composto da  $k$ -simplessi e dalle loro facce (per  $k = 2$ : è composto da triangoli e tutte le loro facce - senza parti staccate...). Una rete di bordi è un complesso simpliciale omogeneo unidimensionale. Una rete di triangoli è un complesso simpliciale omogeneo bidimensionale. Le reti di triangoli sono complessi simpliciali omogenei bidimensionali.

$$\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$$

$$\mathcal{V} = \{v_1, \dots, v_n\} \quad (\text{called vertices})$$

$$\mathcal{E} = \{e_1, \dots, e_n\} \quad e_i = (v_{i1}, v_{i2}) \in \mathcal{V} \times \mathcal{V} \quad (\text{called edges})$$

$$\mathcal{F} = \{f_1, \dots, f_n\} \quad f_i = (v_{i1}, v_{i2}, v_{i3}) \in \mathcal{V} \times \mathcal{V} \times \mathcal{V} \quad (\text{called faces})$$

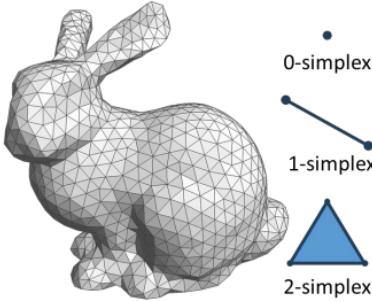


Figure 83: Triangle Meshes

Sulle reti di triangoli, di solito si distinguono gli *aspetti geometrici* (dove sono posizionati i punti nello spazio) dagli aspetti combinatori (come sono connessi gli elementi).

- Due  $k$ -simplessi  $A$  e  $B$  sono  $m$ -adiacenti ( $m < k$ ) se esiste un  $m$ -simplex che è una faccia corretta sia di  $A$  che di  $B$ 
  - Due triangoli che condividono un lato sono 1-adiacenti
  - Due triangoli che condividono un vertice sono 0-adiacenti
  - Due lati che condividono un vertice sono 0-adiacenti
- Due simplessi sono incidenti se uno è una faccia corretta dell'altro
  - Due vertici sono adiacenti se sono incidenti su un lato comune

## 8.2 Star and Link

- **La stella** di un simplesso  $\Delta$  è l'unione di tutti i simplessi che contengono  $\Delta$ .
- **Il link** di  $\Delta$  consiste in tutte le facce dei simplessi nella stella di  $\Delta$  che non intersecano  $\Delta$ .

Per una rete di triangoli, il bordo è l'unione di spigoli incidenti a un solo triangolo.

### 8.2.1 Surface in Computer Graphics

Spesso si assume che siano «varietà continue orientabili bidimensionali immerse in  $\mathbb{R}^3$ ». Intuizione: la superficie di confine di un solido fisico, non degenerato («non degenero» qui significa senza parti infinitamente sottili). Su una superficie *manifold*, i punti hanno vicinanze topologicamente equivalenti (omeomorfe) o a un disco o a un semidisco (se la superficie ha dei confini). Le varietà di dimensione 2 e 3 sono modelli matematici convenienti per la rappresentazione degli oggetti tridimensionali.

- Validità dei modelli digitali (ad esempio, in medicina, per simulazioni, ecc.)
- Strutture dati convenienti (altro da aggiungere nelle prossime lezioni)

### 8.2.2 From continuos to discrete setting

La "manifoldness" si riferisce alla caratteristica principale delle varietà di essere spazi che possono essere descritti, almeno localmente, come spazi euclidei. Per definire la "manifoldness" per le reti di triangoli, abbiamo bisogno di concetti discreti equivalenti di "vicinato" e "disco" (ricordiamo i concetti di stella e link). (Ricordiamo la definizione nel caso continuo: una superficie è una varietà se ogni punto ha un vicinato che può essere deformato continuamente o in un disco o in un semidisco.)

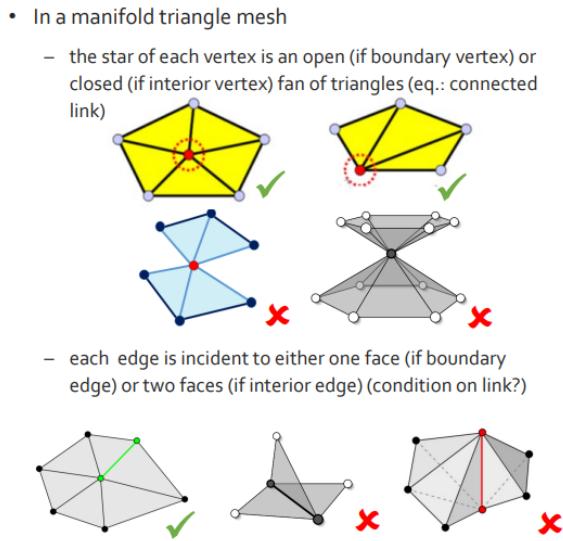


Figure 84: Triangle Meshes

Una varietà combinatoria di dimensione 2 è un complesso omogeneo nel quale ogni vertice è regolare. Si verifica inoltre che tutti i lati siano regolari

## 8.3 Orientabilità

### 8.3.1 Tangent and normal vectors con curves

Una curva piana parametrizzata e liscia può essere rappresentata come una mappa differenziabile da un intervallo nello spazio dei numeri reali allo spazio euclideo bidimensionale. Il vettore tangente (detto anche vettore velocità) alla curva in un punto è la prima derivata della mappa. Esso fornisce la direzione e la velocità del movimento lungo la curva. Il vettore normale viene ottenuto ruotando il vettore tangente di 90 gradi. Esso fornisce la variazione (direzione e magnitudine) nella direzione tangente mentre ci muoviamo lungo la curva. Il **piano tangente** alla superficie in un punto  $p$  è generato dai vettori tangenti alle curve che passano attraverso  $p$ . Il **vettore normale unitario alla superficie** in  $p$  è il vettore ortogonale al piano tangente in  $p$ . Una superficie è **orientabile** se è possibile orientare in modo coerente i normali in ogni punto (ovvero, è a due facce). In caso contrario, è non orientabile (ovvero, è a una faccia).

Esempi ben noti di superfici non orientabili sono il nastro di Möbius e la bottiglia di Klein.

### 8.3.2 Normals on triangle meshes

Molte operazioni richiedono vettori normali (ad esempio, lo shading). Idea: calcolare i normali ai vertici come media pesata dei vettori normali dei triangoli nella stella (nota: i triangoli sono piani, quindi i normali sono costanti e possono essere calcolati come il prodotto vettoriale di due lati).

Pesi costanti possono dare risultati controintuitivi su reti irregolari. Il ponderamento, ad esempio, in base all'area dei triangoli o agli angoli, può essere di aiuto.

## 8.4 Curvature

La **curvatura** è una misura che esprime quanto una superficie o una curva si discosta da essere piatta o dritta. La grandezza del vettore normale è chiamata curvatura. Misura quanto la curva si piega, cioè quanto fortemente la curva si discosta da una linea retta. La curvatura sarà zero per una linea retta. Su una curva planare, la **curvatura** è definita come l'inverso del raggio del cerchio osculatore (il cerchio che approssima al meglio localmente la curva, tangente alla curva; linea retta; cerchio di raggio infinito, di conseguenza...).

### 8.4.1 Normal curvature on surfaces

Per estendere la nozione di curvatura dalle curve alle superfici, definiamo la curvatura normale in  $p$  come la curvatura della curva planare creata dall'intersezione della superficie in  $p$  con il piano generato dal vettore tangente  $t$  e il vettore normale  $n$  della superficie.

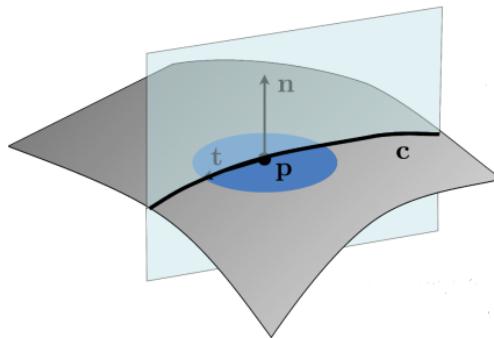


Figure 85: Curvature Surface

La curvatura normale è segnata: è considerata positiva se la curva gira nella stessa direzione del vettore normale della superficie, altrimenti è considerata negativa. La curvatura normale ha due valori estremali distinti (massimo e minimo), chiamati **curvature principali**. Questi valori misurano la massima e la minima curvatura della superficie. Le curvature principali identificano due vettori tangenti unici, chiamati **vettori principali**, che sono sempre ortogonali l'uno all'altro.

### 8.4.2 Mean and Gaussian curvature

Dalle curvature principali possiamo definire due importanti grandezze: la curvatura media e la curvatura gaussiana. Intuizione: la curvatura media come operazione logica OR (c'è curvatura almeno lungo una direzione? Ma attenzione a uguale magnitudine e segno opposto) e la curvatura gaussiana come operazione logica AND (c'è curvatura lungo entrambe le direzioni?). La **curvatura media** è la media delle curvature principali. Una superficie che ha ovunque curvatura media uguale a zero è una superficie minima (ad esempio, pellicole di sapone). La **curvatura gaussiana** è il prodotto delle curvature principali. Il segno della curvatura gaussiana può essere utilizzato per classificare i punti della superficie in tre categorie distinte: ellittiche, iperboliche e paraboliche. La curvatura gaussiana è spesso utilizzata per l'ispezione visiva al fine di valutare la qualità delle superfici nel design industriale: l'estetica di un oggetto dipende fortemente da come riflette la luce, quindi dalle sue proprietà basate sulla curvatura

*mean curvature*

$$H = \frac{\kappa_1 + \kappa_2}{2}$$

*Gaussian curvature*

$$K = \kappa_1 \cdot \kappa_2$$

Figure 86: Mean and Gaussian curvature

## 8.5 Geodesics

Nello spazio euclideo in cui viviamo, la distanza tra due punti è la lunghezza della linea retta che li collega. Le distanze geodetiche generalizzano questa nozione alle superfici: esse misurano i percorsi più brevi tra due punti, se ci si limita a viaggiare sulla superficie stessa. Le distanze geodetiche sono grandezze intrinseche, poiché non dipendono da come la superficie è inserita nello spazio (possono essere calcolate da creature bidimensionali che vivono sulla superficie senza conoscenza della terza dimensione). Le distanze geodetiche sono invariate rispetto alle deformazioni della curvatura. Le mappe che conservano le distanze geodetiche sono chiamate isometrie. **Teorema Egregium:** Afferma che la curvatura gaussiana è una proprietà intrinseca: può essere calcolata da creature bidimensionali che vivono sulla superficie senza conoscenza della terza dimensione. (Risultato sorprendente! Le curvature principali NON sono intrinseche). La curvatura gaussiana è invariante sotto isometrie locali. Una conseguenza del Teorema Egregio è che non esiste alcuna **isometria** tra una sfera (una superficie di curvatura

non nulla) e un piano (una superficie di curvatura nulla). Pertanto, tutte le proiezioni cartografiche distortono la superficie in qualche modo.

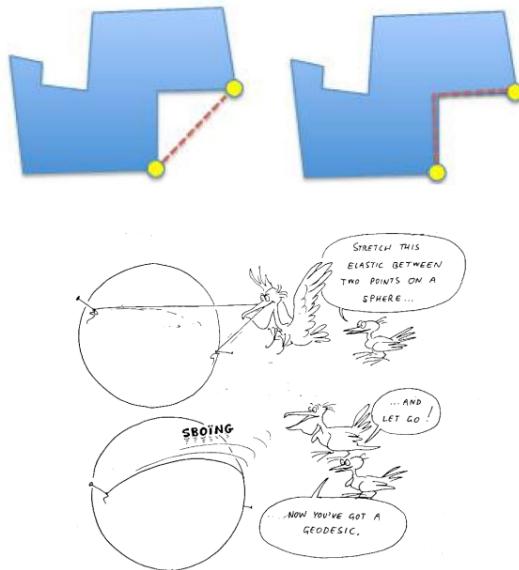


Figure 87: Geodesics distance

## 8.6 Genus

Il **genere** di una superficie连nessa e orientabile è un numero intero che conta il massimo numero di tagli lungo curve semplici chiuse non intersecanti senza disconnettere la superficie. (Abbiamo una definizione più complessa per le superfici non orientabili.) Il genere è un invariante topologico: se due superfici orientabili hanno lo stesso genere, sono topologicamente equivalenti, il che significa che possono essere deformate continuamente l'una nell'altra (senza strappare né incollare).

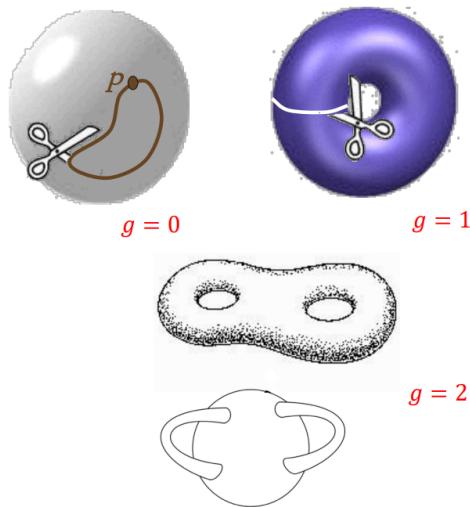


Figure 88: Genus

La caratteristica di Eulero è definita come  $\chi = V - E + F$  dove  $V$  è il numero di vertici,  $E$  è il numero di spigoli,  $F$  è il numero di facce. Si verifica che  $\chi = 2 - 2g - b$  dove  $g$  è il genere della superficie e  $b$  è il numero di bordi.

# 9 Geometric Transformations

Il processo che va dagli oggetti tridimensionali alle scene sintetiche bidimensionali coinvolge una sequenza di trasformazioni da uno spazio all'altro. A sua volta, questa sequenza richiede la definizione e il calcolo di adeguate trasformazioni. L'**object space** è il riferimento in cui si definisce un modello 3D (ad esempio, lo spazio in cui si definiscono le coordinate dei punti, le normali, ...). Ogni oggetto 3D ha il suo spazio, scelto arbitrariamente dal modellatore (ci sono convenzioni). Una scena è di solito composta da diversi oggetti. Il **world space** è il sistema di riferimento comune alla scena. Abbiamo bisogno di una mappa dall'object space al world space: la **trasformazione di modellazione**.

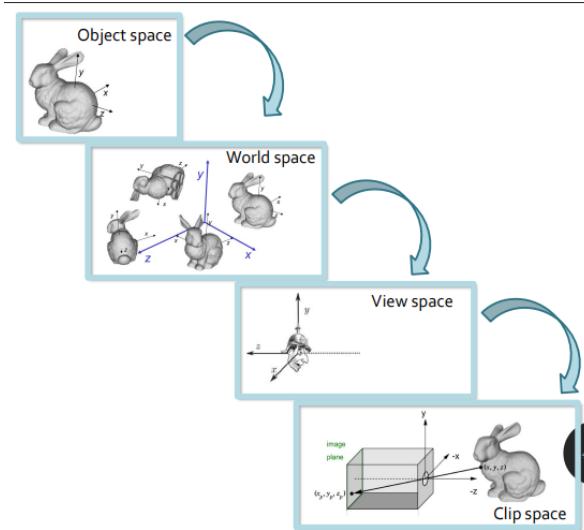


Figure 89: Geometric Transformations

La trasformazione di modellazione riflette come ogni modello è posizionato nella scena. Ogni oggetto 3D ha la propria trasformazione di modellazione. Le trasformazioni includono, ad esempio, rotazioni degli oggetti, traslazioni e scalature. Queste trasformazioni affini sono rappresentate come matrici. Il **view space** (spazio della fotocamera, spazio dell'occhio) ha l'origine nel centro della fotocamera. Abbiamo bisogno di una seconda trasformazione dallo spazio del mondo allo spazio di vista. Nota: le trasformazioni possono essere composte in una singola trasformazione (da oggetto a vista). Lo spazio di clip è lo spazio finale di destinazione, allineato con lo schermo bidimensionale del dispositivo di output. Il mapping dallo spazio di vista allo **spazio di clip** coinvolge la proiezione degli elementi tridimensionali sul piano bidimensionale. Le proiezioni non sono trasformazioni affini. Messaggio chiave: Se vogliamo passare dalla geometria tridimensionale allo schermo, abbiamo bisogno di trasformazioni che agiscano su punti e vettori.

## 9.1 Recap: Point and Vectors

Le entità fondamentali nella Grafica Computazionale sono i **punti** (cioè, le posizioni) e i **vettori** (cioè, gli spostamenti, le direzioni). Anche se hanno rappresentazioni simili

nello spazio tridimensionale (cioè, triplette di coordinate), il loro significato geometrico e fisico è completamente diverso. Operazioni diverse hanno senso per entità diverse.

- point - point = vettore (spostamento)
- point + vector = punto (traslato)
- vector + vector = vettore (legge del parallelogramma)
- vector - vector = vettore (legge del parallelogramma)
- scalare  $\cdot$  vettore = vettore (magnitudine e lunghezza scalate)

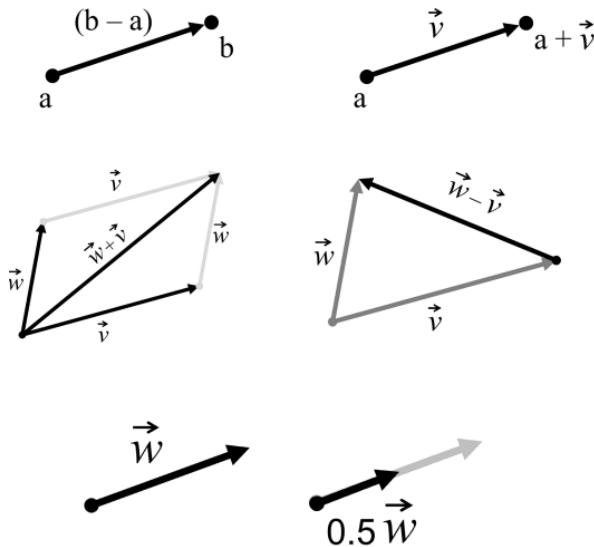


Figure 90: Punti e Vettori

**Prodotto scalare:** da due vettori a uno scalare (somma dei prodotti delle coordinate corrispondenti) **Prodotto vettoriale:** da due vettori a un vettore, ortogonale ad entrambi i vettori di partenza (in  $\mathbb{R}^3$ ); l'orientamento dipende dall'ordine degli operandi.

## 9.2 Affine transformations 2D

Le trasformazioni affini includono **traslazione, rotazione, scalatura, scorrimento e riflessione**. La composizione delle trasformazioni affini è una trasformazione affine.

### 9.2.1 Translation(2D)

Traslare un oggetto geometrico nel piano 2D implica spostare ciascun punto P dell'oggetto lungo l'asse x e l'asse y, tramite due quantità scalari (ricordando l'algebra dei punti e dei vettori: è possibile sommare un punto e un vettore e ottenere...).

### 9.2.2 Scaling(2D)

Rispetto a un punto fisso, qui si assume che il punto sia l'origine del riferimento. Ridimensionare un punto P significa posizionare P ad una nuova distanza dagli assi; ridi-

mentionare un oggetto significa ridimensionare tutti i suoi punti. Effetto: cambiano le dimensioni lungo gli assi (confronta gli effetti dei fattori di ridimensionamento maggiori o minori di 1). Nota: moltiplicazione di matrici.

### 9.2.3 Rotation(2D)

Attorno a un punto fisso Presupposti: centro di rotazione nell'origine del sistema di riferimento Ruotare un punto P significa spostare P attorno al punto fisso per un certo angolo; ancora una volta, ruotare un oggetto significa ruotare solidamente tutti i suoi punti. Angoli positivi implicano direzione antioraria, angoli negativi direzione oraria. Nota: moltiplicazione di matrici. Le formule possono essere calcolate utilizzando l'espressione di P nelle coordinate polari.

### 9.2.4 Composing Transformations

Le formule che abbiamo visto per la rotazione e la scalatura assumono che il punto fisso sia lo stesso dell'origine del sistema di riferimento. Come possiamo fare per ruotare (o ridimensionare) rispetto a un punto che non è l'origine del sistema di riferimento? Idea di composizione delle trasformazioni:

- Applicare una traslazione per spostare l'origine al punto dato
- Ruotare (o scalare)
- Applicare la traslazione inversa per riportare l'origine

Possiamo farlo con una singola matrice? **Problema:**

- La traslazione è espressa come una somma
- La rotazione (o scalatura) è espressa come un prodotto

**Soluzione:**

- Coordinate omogenee: una rappresentazione diversa per punti e vettori (non più coordinate cartesiane)

### 9.2.5 Homogeneous coordinates (2D)

Un punto P le cui coordinate cartesiane sono  $(x, y)$  viene rappresentato con *coordinate omogenee*.  $(x_h, y_h, w), w \neq 0$   $x = x_h/w; y = y_h/w$  Due triple rappresentano lo stesso punto se una è un multiplo dell'altra. Poiché  $w$  non è zero, possiamo adottare la **forma canonica** come rappresentante della classe di coordinate omogenee che rappresentano lo stesso punto.  $(x, y, w) -> (x/w, y/w, 1)$  Vantaggio: le coordinate omogenee eliminano l'ambiguità tra punti (posizioni) e vettori (direzioni). Le terne con 1 come terza coordinata sono punti, mentre le terne con 0 come terza coordinata possono essere viste come punti all'infinito, quindi direzioni (torneremo su questo argomento).

### 9.2.6 Translation revisited (2D)

Nota: poiché ora abbiamo tre coordinate per punti e vettori, avremo matrici di dimensione 3 per 3. È facile verificare che la concatenazione delle traslazioni mediante moltiplicazione dà una traslazione per una quantità lungo ciascun asse che è la somma delle singole quantità.

$$x' = x + d_x; \quad y' = y + d_y$$

- With homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Concatenation

$$\begin{bmatrix} 1 & 0 & d_{x1} \\ 0 & 1 & d_{y1} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & d_{x2} \\ 0 & 1 & d_{y2} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{x1} + d_{x2} \\ 0 & 1 & d_{y1} + d_{y2} \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 91: Punti e Vettori

### 9.2.7 Scaling revisited (2D)

Rispetto a un punto fisso Ancora una volta, concatenazione tramite moltiplicazione di matrici.

$$x' = x + d_x; \quad y' = y + d_y$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

#### Concatenation

$$\begin{bmatrix} 1 & 0 & d_{x1} \\ 0 & 1 & d_{y1} \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & d_{x2} \\ 0 & 1 & d_{y2} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{x1} + d_{x2} \\ 0 & 1 & d_{y1} + d_{y2} \\ 0 & 0 & 1 \end{bmatrix}$$

### 9.2.8 Scaling Revisited

Rispetto a un punto fisso, ancora una volta, concatenazione tramite moltiplicazione di matrici.

$$x' = s_x * x; \quad y' = s_y * y$$

With homogeneous coordinates

$$x' = x + d_x; \quad y' = y + d_y$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x1} * s_{x2} & 0 & 0 \\ 0 & s_{y1} * s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling attorno ad un **punto arbitrario**  $P = (P_x, P_y)$

$$S_g = \begin{bmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -P_x \\ 0 & 1 & -P_y \\ 0 & 0 & 1 \end{bmatrix}$$

### 9.2.9 Rotation revisited (2D)

Intorno a un punto fisso, con il centro di rotazione nell'origine del sistema di riferimento.

$$x' = x * \cos\theta - y * \sin\theta; y' = x * \sin\theta + y * \cos\theta$$

Con coordinate omogenee:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotazione attorno ad un punto arbitrario  $P = (P_x, P_y)$

$$R_g = \begin{bmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -P_x \\ 0 & 1 & -P_y \\ 0 & 0 & 1 \end{bmatrix}$$

### 9.2.10 Riassumendo

Dalle coordinate cartesiane alle coordinate omogenee • Possiamo rappresentare in modo univoco le trasformazioni come matrici, e possiamo facilmente comporre trasformazioni dello stesso tipo • Torniamo al nostro problema di comporre trasformazioni di tipi diversi • La buona notizia è che con le coordinate omogenee, ogni sequenza di trasformazioni si riduce all'applicazione di una singola matrice, che è il prodotto delle singole matrici che rappresentano le trasformazioni (ricorda che questo non era il caso con le coordinate cartesiane)

### 9.2.11 Concatenation of transformations

Nota: l'ordine è importante! La moltiplicazione tra matrici in generale non è commutativa. Esempio: il bianco è l'input, il grigio scuro è il risultato. A sinistra: prima rotazione, poi scalatura; A destra: prima scalatura, poi rotazione.

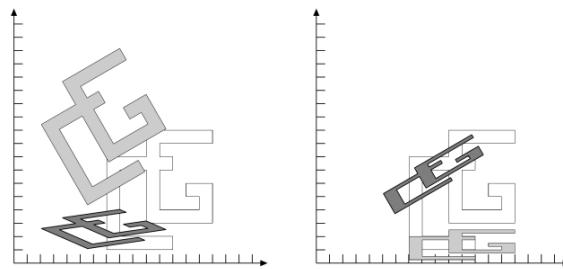


Figure 92: Concatenation of Transformations

Per mappare dallo spazio dell'oggetto allo spazio del mondo, è possibile concatenare trasformazioni affini e rappresentare la concatenazione con una singola matrice, utilizzando coordinate omogenee. Attraverso la moltiplicazione delle matrici, si otterranno le

coordinate nello spazio del mondo. allo stesso modo, le composizioni vengono utilizzate per propagare le trasformazioni lungo i nodi dei grafi della scena.

### 9.2.12 Reflection (2D)

- Reflection with respect to x axis

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Reflection with respect to y axis

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Reflection with respect to the origin

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Figure 93: Reflection(2D)

### 9.2.13 Shear (2D)

La quantità di deformazione di un punto lungo l'asse x dipende dalla coordinata y del punto, e viceversa.

- Shear with respect to x axis

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Reflection with respect to y axis

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Reflection with respect to both axes

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = x + ay \quad y' = y + bx$$

Figure 94: Reflection(2D)

## 9.3 Affine transformations 3D

Si possono utilizzare le coordinate omogenee anche in tre dimensioni.  $p = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

$v = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$  In questo caso abbiamo tre assi utilizzati per definire l'orientamento. Esempio di coordinate omogenee:

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 9.3.1 Rotation(3D)

Abbiamo bisogno di identificare l'angolo, la direzione e l'asse. Le rotazioni intorno agli assi cartesiani possono essere facilmente derivate dal caso 2D. Le rotazioni intorno ad assi arbitrari possono essere ottenute componendo traslazioni + rotazioni intorno agli assi cartesiani.

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 95: Rotation Matrix (3D)

### 9.3.2 Recap from the rendering pipeline

Descriviamo la relazione tra l'osservatore e la fotocamera attraverso la metafora della fotocamera a foro stenopeico. Abbiamo un foro infinitamente piccolo, l'immagine si

forma sul piano sul retro della fotocamera, mentre assumiamo di avere anche un piano virtuale ad una distanza  $d$  dal centro di proiezione, dove  $d$  è la lunghezza focale. Problema: come derivare le coordinate 2D sull'immagine delle entità nella scena 3D. Abbiamo diverse scelte, in base al tipo di informazione che si desidera rappresentare.

### 9.3.3 Prospective and Parallel projections

Le proiezioni sono definite da una serie di linee rette, chiamate proiettori, che hanno un'origine comune nel centro di proiezione. Attraversano i punti della scena 3D e intersecano il piano di proiezione per formare la proiezione. Poiché la proiezione di un segmento è un segmento, proiettare i vertici è sufficiente. Se i proiettori sono linee rette e il piano di proiezione è effettivamente un piano, parliamo di proiezioni planari. **Proiezioni prospettiche**: il centro ha una distanza finita dal piano. **Proiezioni parallele**: la distanza non è finita e i proiettori sono linee parallele. Effetto fisheye vs preservare le proporzioni.

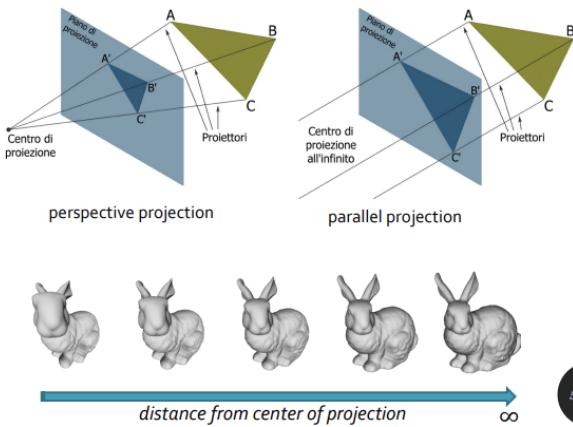


Figure 96: Prospective and Parallel

Le proiezioni prospettiche sono realistiche. Tuttavia, le distanze non vengono preservate e le linee parallele non sono più parallele. Con le proiezioni parallele, invece, la vista può essere meno realistica, ma le distanze vengono preservate così come il parallelismo, quindi sono utili se si devono effettuare misurazioni.

Proiezioni ortografiche parallele: il piano dell'immagine è perpendicolare a uno degli assi del sistema cartesiano, che identifica la direzione di proiezione.

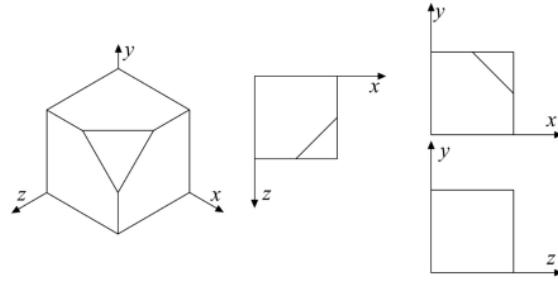


Figure 97: Prospective and Parallel

### 9.3.4 Perspective projection matrix

$$M_{pro} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}$$

$$\begin{aligned} \frac{x_p}{d} &= \frac{x}{z}; \frac{y_p}{d} = \frac{y}{z}, \\ x_p &= \frac{x}{z/d}, y_p = \frac{y}{z/d}, z_p = d \\ (x_p, y_p, z_p) &= \left( \frac{x}{z/d}, \frac{y}{z/d}, d \right) M_{pro} * P = (X; Y; Z; W) = (x, y, z, \frac{z}{d}) \\ \left( \frac{X}{W}, \frac{Y}{W}, \frac{Z}{W} \right) &= (x_p, y_p, z_p) = \left( \frac{x}{z/d}, \frac{y}{z/d}, d \right) \end{aligned}$$

### 9.3.5 Parallel projection matrices

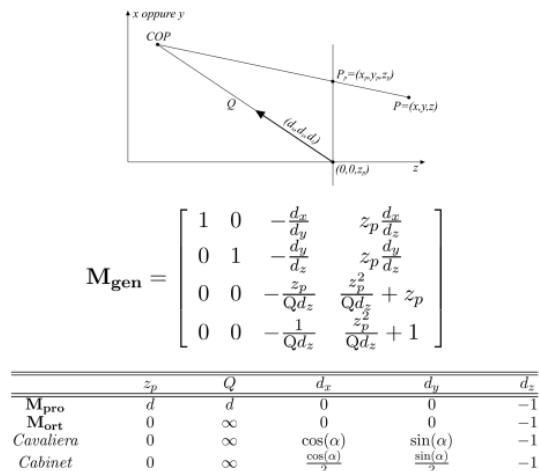


Figure 98: Prospective projection matrix

$$M_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# 10 Introduction to Rendering

Il **Rendering** è il processo che converte i dati in input in immagini. Paradigmi del Rendering:

- Ray Tracing
- Path Tracing
- Rasterization-based Pipeline

per visualizzare scene 3D abbiamo bisogno di trasformare le immagini sintetiche che possono essere mostrate sullo schermo. Esistono diversi algoritmi per trasformare una scena 3D in una raster image, in seguito verranno presentati due algoritmi appartenenti a due famiglie differenti:

- Ray tracing algorithms
- Rasterization-based algorithms

## 10.1 Pinhole Camera

La metafora usata per descrivere la relazione tra l'osservatore e la scena è quella della camera virtuale. Per convenzione, assumiamo l'esistenza di un piano immagine tra la scena e il centro di proiezione.

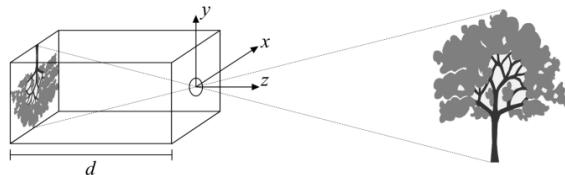


Figure 99: Pinhole camera

La fotocamera virtuale consiste in un parallelepipedo nel quale la faccia anteriore ha un foro infinitesimale (fotocamera a foro stenopeico) e sul retro vengono formate le immagini; l'angolo  $\theta$  di visualizzazione può essere modificato variando il rapporto tra la distanza focale  $d$  e la dimensione  $h$  del piano immagine.

### 10.1.1 Perspective Projection

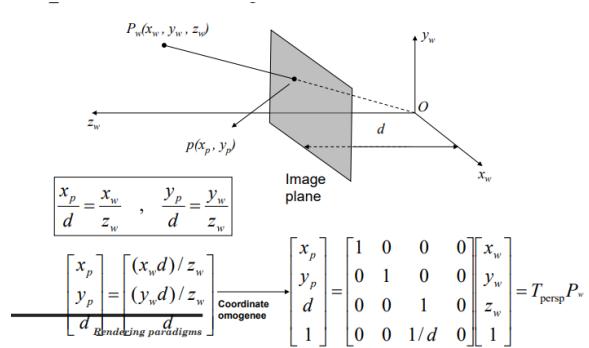


Figure 100: Prospective Projection

così possiamo scrivere

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

o in maniera equivalente:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K[I|0]P_w \quad K = \begin{bmatrix} d & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix} P_w$$

### 10.1.2 From image plane to pixel coordinates

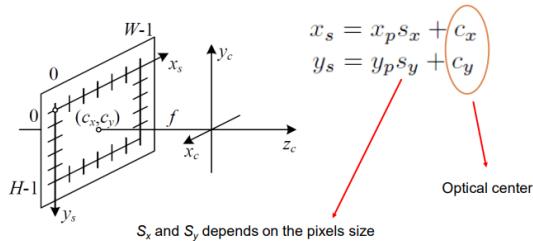


Figure 101: To pixel coordinates

### 10.1.3 Camera Model

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K[I|0]P_w$$

Calibration matrix, von  $f_{sx}$  ey  $f_{sy}$  *facal lenght*:

$$K = \begin{bmatrix} f_{sx} & s & c_x \\ 0 & f_{sy} & c_y \\ 0 & 0 & 1 \end{bmatrix} P_w$$

Versione Usuale:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} P_w$$

Prendendo in considerazione la posizione e l'orientamento della camera nella scena:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K[I|0]MP_w \quad M = \begin{bmatrix} R & t \\ 1 & 0 \end{bmatrix}$$

*p\_s p\_i o meno PP\_w*

Quindi, abbiamo due insiemi di parametri che caratterizzano la fotocamera.

- **Parametri intrinseci:** lunghezza focale, dimensione dei pixel, risoluzione, centro ottico.
- **Parametri estrinseci:** posizione e orientamento della fotocamera, cioè una rotazione e una traslazione.

## 10.2 Ray-Tracing

Idea principale: tracciare i raggi luminosi all'indietro Per ogni pixel dello schermo:

- Sparare un raggio (chiamato raggio primario)
- Trovare l'intersezione con la primitiva più vicina.

Il nucleo è: l'intersezione raggio/primitiva (deve essere estremamente ottimizzata). Facile da gestire: riflessioni speculari multiple, facile da realizzare: semitrasparenza con rifrazioni, facile da gestire: ombre (nitide).

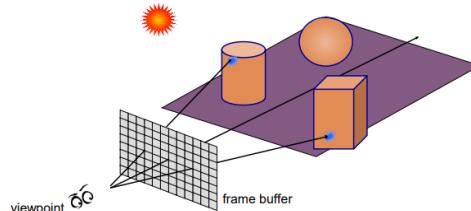


Figure 102: Ray-Tracing

## Pseudocodice

```
Result: Color the pixels on the screen
for each pixel in the screen do
    Generate a primary ray from the camera's viewpoint through the pixel;
    Find the closest intersection between the ray and objects in the scene;
    if an intersection exists then
        Calculate the color at the intersection point considering material
        properties, lights, and illumination;
        If needed, cast further rays (secondary rays) for effects like shadows,
        reflections, refractions;
        Combine the computed colors via secondary ray intersections with the
        main color;
        Assign the resulting color to the pixel on the screen;
    end
    else
        | Assign a background color to the pixel;
    end
end
```

**Algorithm 1:** Pseudocode for Basic Ray Tracing



Figure 103: Esempio di Ray-Tracing

- Nucleo principale:
  - Calcolo dell'intersezione tra un **RAGGIO 3D** e **PRIMITIVI 3D**
- Computazionalmente complesso:
  - Elevato costo
  - Tuttavia SUBLINEARE! (rispetto al conteggio delle primitive)
- Se si utilizzano le strutture corrette: **indicizzazione spaziale!**
- L'indicizzazione spaziale sono un insieme di strutture dati speciali (gerarchiche, basate sull'hashing) per accelerare il calcolo dell'intersezione raggio/scena.
- Una volta utilizzate solo per elaborazioni offline:
  - Ora in tempo reale nei casi più semplici sfruttando le GPU

### 10.3 Rasterization-Based Rendering

La rasterizzazione è il processo di conversione di oggetti o immagini descritte in forma vettoriale o geometrica in una griglia regolare di pixel su uno schermo o un dispositivo di visualizzazione. Questo processo è utilizzato per rappresentare graficamente oggetti o immagini su schermi bidimensionali, come monitor, televisori o dispositivi mobili.

In sostanza, la rasterizzazione prende informazioni continue o vettoriali e le traduce in informazioni discrete su una griglia di pixel. Ad esempio, in grafica 3D, gli oggetti tridimensionali vengono proiettati su una vista bidimensionale e successivamente convertiti in pixel attraverso la rasterizzazione per essere visualizzati su uno schermo.

```

foreach pixel  $p$  do
    Make a ray  $r$ ;
    foreach primitive  $o$  in
        scene do
            if  $\text{intersect}(r, o)$  then
                Find color for  $o$ ;
                Color  $p$  with it;
            end
        end
    end
end

foreach primitive  $o$  do
    Find where  $o$  falls on screen;
    Rasterize 2D shape;
    foreach produced pixel  $p$  do
        Find color for  $o$ ;
        Color  $p$  with it;
    end
end

```

**Algorithm 2:** Algorithm 1: Ray-Tracing

**Trasformazioni:**

- Trasformazioni dei sistemi di coordinate
- Obiettivo: proiettare le primitive nello spazio dello schermo
- ... ma anche assemblare le primitive per comporre la scena

**Illuminazione:**

- Calcolo dell'illuminazione
- Obiettivo: calcolare il colore finale di ogni parte della scena
- Per fare ciò dobbiamo considerare:
  - Caratteristiche ottiche
  - Illuminazione della scena
  - Geometria degli oggetti

Dato che l'hardware grafico è ottimizzato per rasterizzare triangoli, l'intera scena viene decomposta in TRIANGOLI tridimensionali

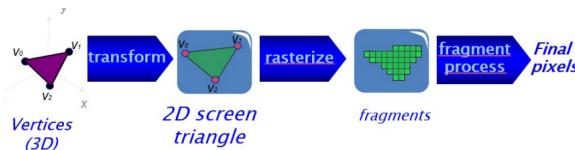


Figure 104: Rasterization-Based Rendering

Input: insieme di vertici e i loro attributi associati

- L'algoritmo passa attraverso diverse fasi:
  1. Trasformazioni per vertice e configurazione degli attributi
  2. Elaborazione delle primitive
  3. Rasterizzazione
  4. Calcolo per frammento

## 10.4 (

Rasterization vs Ray-Tracing)

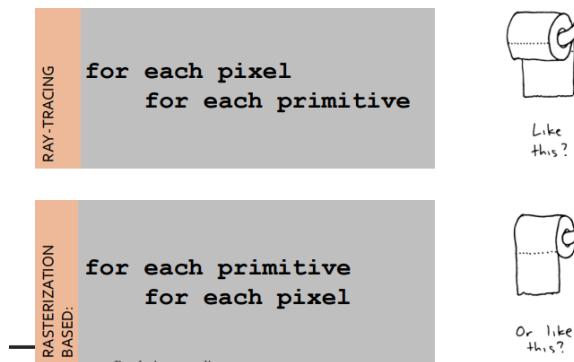


Figure 105: Rendering Algorithms Paradigms

Vantaggi del **Ray-Tracing**:

- buono per complessi effetti visuali
- Algoritmo concettualmente semplice
- Considera gli effetti GLOBALI
- Rappresentazione più realistica degli effetti luminosi
- Maggiore libertà delle primitive (è più semplice intersecare che proiettare e rasterizzare)
- Potenzialmente grande scalabilità con la complessità della scena

Vantaggi della **Rasterization**:

- veloce, per ogni primitiva vengono processati solo alcuni pixels!
- Più facilmente parallelizzabile?
- Complessità più controllabile
- Più adatto alle schede grafiche (per ora)

- Più semplice con dati dinamici?
- Migliore gestione dell'anti-aliasing (più avanti)
- Migliore scalabilità con la risoluzione dell'immagine

#### 10.4.1 Visione Storica

##### **Ray-tracing**

- Algoritmo di scelta per il rendering OFFLINE
- Utilizzato per FILM (e immagini statiche CGI)
- Preferito da Pixar, Dreamworks, ecc.
- Noti ray-tracer: POV-ray, RenderMan, YafaRay...

##### **Rasterization**

- Algoritmo di scelta per il rendering in TEMPO REALE
- Utilizzato per GIOCHI (e anteprime, e 3D sul web...)
- Preferito da Nvidia, ATI, ecc.
- Primo a ottenere hardware specializzato

#### 10.4.2 Reality

Non necessariamente LENTO!

- Algoritmi e strutture dati per "query spaziali" efficienti
  - Anche sub-lineare rispetto al numero di primitive!
- .. non necessariamente SOLO software:
- Hardware specializzato?
  - Progetto OpenRT
  - Tecnologie di tracciamento dei raggi in tempo reale inTrace GmbH
  - MPI Informatik, Saarbruecken - Ingo Wald 2004 Paradigmi di rendering 50
  - ..non necessariamente priva di effetti visivi complessi (anzi, richiedono solo alcune approssimazioni e alcuni algoritmi avanzati)

# 11 Texturing

Il concetto di **texture** è molto importante nella Grafica Computazionale

- Texturing → modulare un attributo dei vertici per ottenere un effetto visivo
- Possibili attributi dei vertici: colore, normale, trasparenza, un parametro dell'equazione di illuminazione, ecc.
- L'attributo più immediato per trasmettere dettagli visivi alla superficie è il colore.
- La modulazione dell'attributo colore è chiamata texture mapping

Durante le operazioni sui frammenti è possibile accedere a una particolare RAM, chiamata *texture RAM*, che contiene un insieme di texture. Ogni texture è un array 1D, 2D o 3D di texel (**texel** = campione di texture, **pixel** = campione di immagine) dello stesso tipo di dati.

## 11.1 Texels

- Un texel è un colore (componenti: R-G-B o R-G-B-A): la texture è una mappa dei colori
- Un texel rappresenta il componente alpha: la texture è una mappa alpha
- Un texel è una normale (componenti: X-Y-Z): la texture è una mappa delle normali
- Un texel è un coefficiente speculare: la texture è una mappa di brillantezza



Figure 106: Esempio di processo di texturing

Il texture space è anche chiamato spazio uv, le coordinate nello spazio uv sono assegnate ad ogni vertice.

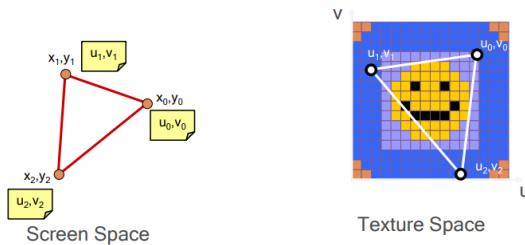


Figure 107: Texturing mapping

Questo definisce un mapping tra i triangoli e le texture.

## 11.2 Attribute Interpolation & Coordinare Baricentriche

Le coordinate baricentriche sono utilizzate

- Cosa sono le coordinate baricentriche?
- Ricorda che un segmento può essere scritto come combinazione lineare di due punti:
- $x = av_1 + bv_2$   
 $a \text{ and } b \text{ positive scalars with } a + b = 1$   
 $(\Rightarrow 0 < a \leq 1 \text{ and } 0 \leq b \leq 1)$

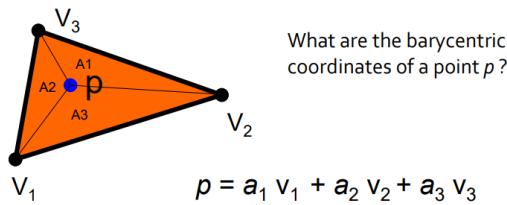


Figure 108: Barycentric Coordinates

Un trinagolo di vertici  $v_1, v_2, v_3$  sono un insieme di punti tali che:

$$x = a_1 v_1 + a_2 v_2 + a_3 v_3$$

$a_1, a_2, a_3$  scalari positivi

$$a_1 + a_2 + a_3 = 1$$

L'interpolazione delle coordinate baricentriche non funziona quando le coordinate della texture sono interpolate a causa della proiezione prospettica. Funziona per altri attributi (ad esempio, per le normali).

### 11.2.1 interpolazione delle coordinate della texture

$$p = c_0 v_0 + c_1 v_1 + c_2 v_2$$

**Attributi** di P:

(senza la Perspective correction)

$$A_p = c_0 A_0 + c_1 A_1 + c_2 A_2$$

$$B_p = c_0 B_0 + c_1 B_1 + c_2 B_2$$

$$p = c_0 v_0 + c_1 v_1 + c_2 v_2$$

(con la Perspective correction)

$$A_p = \frac{c_0 \frac{A_0}{w_0} + c_1 \frac{A_1}{w_1} + c_2 \frac{A_2}{w_2}}{c_0 \frac{1}{w_0} + c_1 \frac{1}{w_1} + c_2 \frac{1}{w_2}}$$

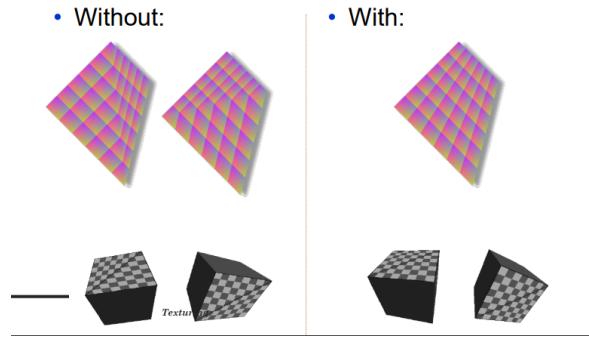


Figure 109: Perspective Correction

Dalla RAM della CPU, la texture deve essere copiata nella RAM della texture per poter essere utilizzata. L'inverso è possibile. Entrambe le operazioni sono lente, quindi è importante progettare la tua applicazione grafica per ottimizzare i trasferimenti delle texture. Come assegnare le coordinate della texture?

- Calcolare le coordinate della texture al volo durante il rendering
- Precomputarle e memorizzarle nella mesh
- Spesso assegnate durante la fase di modellazione

Non esiste una soluzione ideale, dipende dall'applicazione.

### 11.3 UV Mapping

UV Mapping: il problema di assegnare le coordinate della texture a ciascun vertice della mesh come passaggio di pre-elaborazione. Usiamo una funzione di proiezione per fare il mapping dalle coordinate (x,y,z) a quelle (u,v) Planar projection(asse x):

$$(x, y, z) \rightarrow (u, v) : \begin{cases} u = z \\ v = y \end{cases} \quad \text{Cylindric projection:}$$

$$(x, y, z) \rightarrow (u, v) : \begin{cases} u = \text{atan}(z, x) \\ v = y \end{cases}$$

- Cylindric coordinates → Cartesian coordinates
 
$$\begin{aligned} x &= \rho \cos \phi \\ y &= y \\ z &= \rho \sin \phi \end{aligned}$$
- Cartesian coordinates → Cylindric coordinates
 
$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} \\ y &= y \\ \phi &= \arctan(z/x) \end{aligned}$$

*Atan2(z,x) → arctangent of z/x. It returns a value in the  $[-\pi, \pi]$  range by analyzing the sign of z and x.*

Figure 110: Cylindric Projection

### 11.3.1 Enviroment mapping: Sphere mapping

Mappatura dell'ambiente: una texture che rappresenta il colore dell'ambiente riflesso.  
Come coordinate UV può essere utilizzato il vettore di riflessione.  $u = \text{atan}2(r_y, r_x)$   
 $v = \arccos(r_z)$

$$\begin{aligned}\text{Coordinate sferiche -> Coordinate cartesiane } &x = \rho \sin\theta \cos \phi \\ &y = \rho \cos \theta\end{aligned}$$

$$z = \rho \sin \theta \sin \phi$$

$$\text{Coordinate cartesiane -> Coordinate sferiche } \rho = \sqrt{x^2 + y^2 + z^2} \quad \phi = \arctan(z/x)$$

$$\theta = \arctan\left(\frac{y}{x^2+y^2+z^2}\right)$$

### 11.3.2 Enviroment mapping: Cube mapping

- Campionamento più uniforme della mappatura sferica.
- È facile generare la mappa dell'ambiente.
- Generazione delle coordinate uv a partire dal vettore di riflessione ( $r = (rx, ry, rz)$ ):
  - Il valore massimo definisce la faccia del cubo da utilizzare per la mappatura.
  - Esempio:  $(-3.2, 5.1, -8.4) \rightarrow$  faccia -Z
  - Le coordinate sono ottenute dividendo il valore rimanente per il valore massimo (intervallo  $[-1,1]$ ) e normalizzando i risultati tra  $[0,1]$ .
  - Esempio:  $((-\frac{3.2}{8.4}, \frac{5.1}{8.4}) \rightarrow (-0.38, 0.61))$
  - Per normalizzare i valori nell'intervallo  $[-1, 1]$  a  $[0, 1]$  aggiungiamo 1 e dividiamo per 2.
  - Esempio:  $((\frac{(-0.38+1)}{2}, \frac{(0.61+1)}{2}) \rightarrow (0.31, 0.80))$

### 11.3.3 Texturing out of range

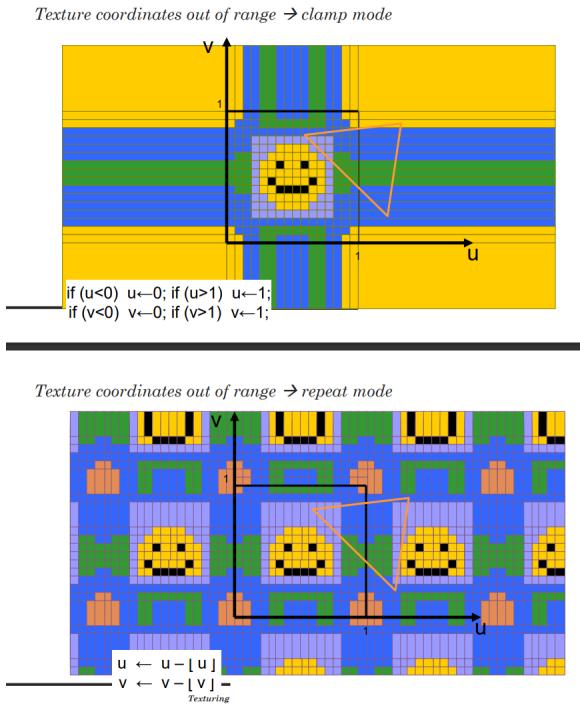
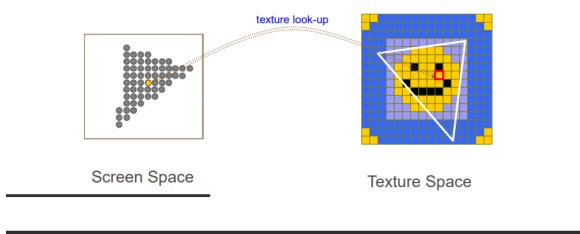


Figure 111: Texture coordinates out of range

## 11.4 Texture Look-Up

### Texture Look-Up

- A fragment has non-integer coordinates (in texels)



### Texture Look-Up

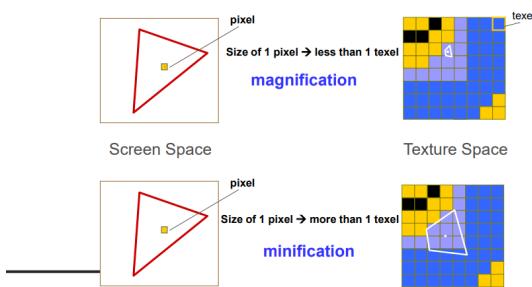


Figure 112: Texture Look-Up

#### 11.4.1 Maginification

Due Soluzioni:

- Ottenere il texel quando cade (è equivalente a ottenere il texel più vicino) È equivalente a arrotondare le coordinate della texture al valore intero.
- Calcolare una media pesata dei quattro texel più vicini. Interpolazione bilineare



Figure 113: Maginification

#### 11.4.2 Minification

#### 11.4.3 Mip-Mapping

Definiamo un fattore di scala  $\rho = \text{texels/pixels}$  come il valore massimo tra  $\rho_x$  e  $\rho_y$ , che può essere derivato dalla matrice di trasformazione geometrica e viene calcolato sui vertici, e interpolato per i frammenti. Il livello del mipmap è:  $\log_2 \rho$ , il livello 0 indica la massima risoluzione. Il livello non è necessariamente un numero intero, quindi deve essere arrotondato.

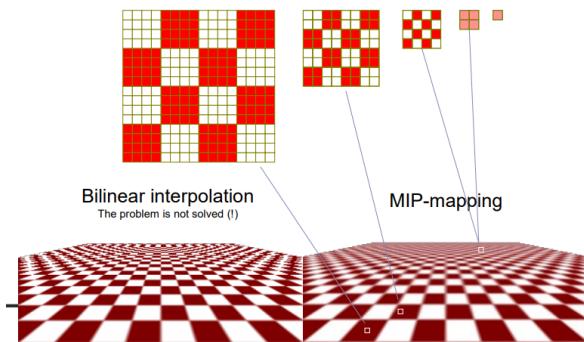


Figure 114: Mip-Mapping & Bilinear

#### 11.4.4 Bump Mapping

L'aspetto di un oggetto può essere modificato utilizzando una tecnica chiamata *Bump Mapping*.

$$\vec{N}_{\text{new}} = \vec{N}_{\text{old}} + \vec{D};$$

Il risultato è una perturbazione delle normali che altera il rendering senza modificare la geometria.

$$\vec{D} = (\Delta x, \Delta y, \Delta z)$$

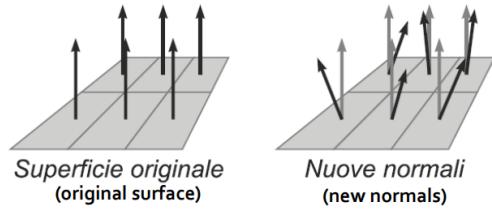


Figure 115: Bump Mapping

#### 11.4.5 Displacement Mapping

Utilizzando il displacement mapping, la geometria dell'oggetto 3D viene effettivamente modificata spostando i punti 3D della superficie:  $P_{new} = P_{old} + h * N$

- Il displacement mapping viene eseguito al momento del rendering e non modifica la geometria della scena.
- La differenza con il bump mapping è che la silhouette del modello 3D è correttamente deformata.

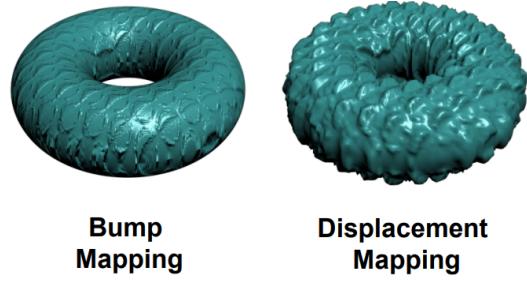


Figure 116: Bump Mapping & displace Mapping

# 12 Lighting

## 12.1 Modelli Matematici della luce

1. **Ottica geometrica:** la luce è modellata utilizzando raggi. I raggi seguono regole geometriche. L'ottica geometrica può modellare diversi effetti luminosi come la riflessione e la rifrazione.
2. **Ottica ondulatoria:** la luce è modellata come un'onda che si propaga. L'ottica ondulatoria può modellare effetti luminosi come la diffrazione.
3. **Ottica elettromagnetica:** può descrivere effetti luminosi come la polarizzazione della luce, non spiegata dall'ottica ondulatoria.
4. **Ottica dei fotoni:** la meccanica quantistica viene utilizzata per modellare gli effetti luminosi.

La luce energetica è emessa da una sorgente di luce, questa viaggia attraverso la scena e interagisce con la scena finché non si stabilizza. Questo processo accade alla velocità della luce. Quindi, per noi, tutto questo processo è stabile  
Cosa succede quando un raggio di luce colpisce una superficie?

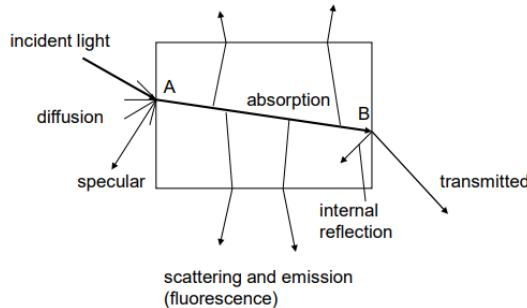


Figure 117: Light-material interaction

## 12.2 Solid angle

Rappresenta la dimensione angolare di un conoide infinitesimale lungo una data direzione. Può essere interpretato come una direzione associata a un'area infinitesimale sulla sfera unitaria (unità di misura: steradiani). È l'estensione nello spazio tridimensionale del concetto di angolo planare.  $\theta$  è misurato in radianti come il rapporto  $\frac{s}{r}$ , dove  $s$  è la lunghezza dell'arco di un cerchio di raggio  $r$  sotteso da  $\theta$ . Analogamente,  $\Omega$  è misurato in steradiani come il rapporto  $\frac{A}{r^2}$ , dove  $A$  è l'area superficiale di una sfera di raggio  $r$  sottesa da  $\Omega$ .

Esempi:

- Angolo retto:  $\frac{2\pi r}{4r} = \frac{\pi}{2}$  radianti
- Angolo solido di un'emisfero:  $\frac{4\pi r^2}{2r^2} = 2\pi$  steradiani

## 12.3 Radiometry

**Flusso radiante** (watt) è l'energia radiante che attraversa una superficie nell'unità di tempo:  $\Phi = \frac{dQ}{dt}$

**Irradianza (E)** (watt/m<sup>2</sup>) è il flusso radiante incidente su un elemento di superficie:  $E(x) = \frac{d\Phi}{dA}$

**La radianza (L)** (watt/m<sup>2</sup>sr) è il flusso radiante per angolo solido per unità di area  $L(x, \vec{w}) = \frac{d^2\phi}{\cos\theta d\omega dA}$

L'irradianza ( $E(x)$ ) è data dall'integrale della radianza incidente ( $L$ ) lungo tutte le direzioni. Il flusso radiante è dato dall'integrale della radianza incidente ( $L$ ) lungo tutte le direzioni e dall'area considerata.  $E(x) = \int_{\Omega} L(x, \vec{w}')(\vec{w}' \cdot \vec{n})d\vec{w}'$

Una sorgente luminosa puntiforme emette uniformemente la luce in tutte le direzioni, producendo:

$$E(x) = \frac{\Phi_s \cos\theta}{4\pi r^2}$$

L'intensità della luce diminuisce con il quadrato della distanza.

## 12.4 e BSSRDF (Bidirectional Scattering Surface Reflectance Distribution Function)

Quando la luce colpisce una superficie interagisce con essa e, nel caso generale, esce dalla superficie da una posizione diversa da quella di ingresso (dispersione). Il **BSSRDF** (Bidirectional Scattering Surface Reflectance Distribution Function) è una funzione che descrive il processo di dispersione. Il BSSRDF può essere scritto come:

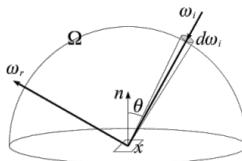
$$S(x_i, \vec{w}_i, x_r, \vec{w}_r) = \frac{dL_r(x_r, \vec{w}_r)}{d\Phi_i(x_i, \vec{w}_i)}$$

Dove  $L_r$  rappresenta la radianza uscente dal punto  $x_r$  nella direzione  $w_r$  e  $\Phi_i$  è il flusso radiante incidente nel punto  $x_i$  dalla direzione  $w_i$ . È una funzione di otto parametri! (due parametri per identificare una posizione sulla superficie e due angoli per specificare una direzione nelle coordinate sferiche).

La BRDF (Bidirectional Reflectance Distribution Function) è un'approssimazione del BSSRDF per descrivere la riflessione della luce da parte di una superficie. In particolare, la BRDF è il rapporto tra la radianza uscente (luce riflessa) in una certa direzione e in un dato punto della superficie ( $x$ ) e l'irradianza incidente nello stesso punto proveniente da una direzione specifica.

$$f_r(x, \vec{w}_i, \vec{w}_r) = \frac{dL_r(x, \vec{w}_r)}{dE_i(x, \vec{w}_i)} = \frac{dL_r(x, \vec{w}_r)}{L_i(x, \vec{w}_i)(\vec{w}_i \cdot \vec{n})d\vec{w}_i}$$

## 12.5 The Rendering Equation



$$L_o(x, \vec{w}_r) = L_e(x, \vec{w}_r) + L_r(x, \vec{w}_r)$$

$$L_o(x, \vec{w}_r) = L_e(x, \vec{w}_r) + \int_{\Omega} f_r(x, \vec{w}_i, \vec{w}_r) L_i(x, \vec{w}_i)(\vec{w}_i \cdot \vec{n})d\vec{w}_i$$

Figure 118: Rendering Equations

La luce visibile in un punto della scena per una direzione specifica è data dalla luce emessa più la luce riflessa in quella direzione.

$$L_o(x, \vec{w}_r) = L_e(x, \vec{w}_r) + L_r(x, \vec{w}_r)$$

La luce riflessa è ottenuta integrando tutte le contribuzioni ponderate in base alla BRDF (che dipende dal materiale specifico) e agli angoli di riflessione  $L_o(x, \vec{w}_r) = L_e(x, \vec{w}_r) + \int_{\Omega} f_r(x, \vec{w}_i, \vec{w}_r) L_i(x, \vec{w}_i)(\vec{w}_i, \vec{n}) d\vec{w}_i$

$x$  Punto sulla superficie dove viene calcolata l'equazione

$\vec{w}_r$  Direzione della radianza riflessa;

$\vec{w}_i$  Direzione del raggio luminoso incidente (radianza in ingresso)

$f_r$  Funzione che determina la frazione della luce riflessa

$\vec{w}_i \cdot \vec{n}$  Coseno dell'angolo di incidenza (rispetto alla normale della superficie)

La computazione dell'Equazione di Rendering è molto dispendiosa È necessario calcolarla diverse volte per molte parti della scena La soluzione è utilizzare un modello di illuminazione locale Semplificazione (approssimazione) Vengono modellati solo gli effetti locali

### 12.5.1 Local and Global effects

Gli effetti visivi causati dalla luce possono essere suddivisi in effetti locali o globali.

- **Modello di illuminazione:** formulazione matematica dell'equazione dell'illuminazione. Tipicamente, i modelli di illuminazione sono approssimazioni dell'Equazione di Rendering.
- **Illuminazione:** calcolo della luce.
- **Shading:** calcolo del colore di ciascun pixel.

### 12.5.2 Phong Lighting model

Creato da Phong Bui-Tuong attorno al 1975, semplifica l'interazione Simula l'interazione luce-materiale con materiali opachi.

- La rifrazione non è modellata: non può essere utilizzata per materiali semitransparenti o trasparenti.
- Piuttosto realistico... Ma alcuni materiali sembrano plastica.
- È composto da tre termini: un termine che modella la componente diffusa del materiale, un termine che modella la componente speculare del materiale e un termine ambientale che cattura le inter-riflessioni della scena.

L'obbiettivo è approssimare il **brdf**, il metodo è quello di semplificare la legge fisica che sta dietro alla specular reflection e la diffuse reflection.

### 12.5.3 Fresnel Law

Quando un raggio di luce passa da un mezzo all'altro con diverso indice di rifrazione, alla superficie di separazione una parte del raggio viene riflesso e una parte viene trasmesso. La somma dell'energia dei due raggi è uguale all'energia del raggio originale. Se non c'è rifrazione dall'aria all'oggetto solido, il raggio viene interamente riflesso. L'angolo di riflessione è uguale all'angolo di incidenza. Questo principio è valido per materiali lisci e lucidi.

### 12.5.4 Lambert Law

I materiali opachi (es. legno, pietra) sono caratterizzati da una superficie con facette microscopiche che riflettono la luce in modo casuale in ogni direzione. A livello macroscopico, la luce viene riflessa uniformemente in tutte le direzioni, con un'intensità proporzionale alla direzione del raggio incidente e alla normale della superficie.

### 12.5.5 Diffuse Reflection

Sorgenti luminose puntiformi sono caratterizzate da:

- La posizione  $P$
- L'intensità della luce emessa  $I_P$

La direzione della luce incidente è:

$$\vec{L} = |L - P|$$

Tenendo conto dell'angolo Illuminazione:

$$\cos(\theta) = \vec{L} \cdot \vec{N}$$

La funzione di riflessione è approssimata come una costante  $k_d$  che dipende dal materiale.

Equazione dell'illuminazione (diffusa):  $I = I_p k_d \cos \theta$

or

$I = I_p k_d \vec{N} \cdot \vec{L}$  Dobbiamo ottenere il massimo tra il coseno e zero nella formula precedente.

$$I = I_p k_d \max(0, \cos \theta) = I_p k_d \cos(0, \vec{N} \cdot \vec{L})$$

### 12.5.6 Specular Reflection

Riflettore non ideale: Un'approssimazione più realistica rispetto alla legge di Fresnel -> evidenza speculare In pratica, l'evidenza speculare è data dalla luce riflessa nella direzione dell'osservatore, il che significa che la sua posizione sull'oggetto dipende dall'osservatore. L'aspetto di un oggetto opaco (riflessione diffusa solamente) non dipende dalla posizione dell'osservatore. C'è una dipendenza dall'angolo tra la direzione di riflessione ideale  $R$  e la direzione della vista  $V$ . Massima riflessione per  $\alpha = 0$  Rapido decadimento quando i valori di alfa aumentano. Questo decadimento è modellato con un esponente ( $n$ ) applicato al coseno dell'angolo  $\alpha$ . Il parametro  $n$  è l'esponente della riflessione speculare del materiale. Il vettore  $R$  è calcolato come:

$$\vec{R} = 2v\vec{N}(\vec{N} \cdot \vec{L}) - \vec{L}$$

L'equazione dell'illuminazione per la riflessione speculare solamente:

$$I = I_p k_s \cos^n \alpha$$

Il parametro  $k_s$  caratterizza il materiale speculare, insieme all'esponente n.

### 12.5.7 Ambient Component

Le inter-riflessioni tra gli oggetti diversi della scena non sono modellate dal modello di Phong. Le inter-riflessioni possono essere approssimate dal seguente componente:

$$I = I_a k_a$$

- $I$  rappresenta il totale dell'energia radiante emessa dalla scena
- $K_a$  modella la riflettività del materiale
- $I_a$  è costante per tutti i punti della scena.

La luce ambientale aggiunge realismo alla scena, anche se è un'approssimazione approssimativa dell'illuminazione indiretta. Tutte le contribuzioni descritte vengono sommate insieme per formare l'equazione dell'illuminazione. Dobbiamo sommare la luce per ciascuna sorgente luminosa presente nella scena.

$$I = \underbrace{I_a k_a}_{\text{Ambient term}} + \sum_p I_p [\underbrace{k_d (\vec{N} \cdot \vec{L})}_{\text{Diffuse term}} + \underbrace{k_s (\vec{R} \cdot \vec{V})^n}_{\text{Specular term}}]$$

Figure 119: Summation

È possibile tenere conto dell'attenuazione della luce dovuta alla distanza.

$$I = I_a k_a + \sum_p f_{att,p} I_p [k_d (\vec{N} \cdot \vec{L}) + k_s (\vec{R} \cdot \vec{V})^n]$$

Considerando una rappresentazione RGB, l'equazione deve essere elaborata separatamente per ogni componente cromatica.

$$\begin{aligned} I_r &= I_{a,r} k_{a,r} + I_{p,r} [k_{d,r} (\vec{N} \cdot \vec{L}) + k_{s,r} (\vec{R} \cdot \vec{V})^n] \\ I_g &= I_{a,g} k_{a,g} + I_{p,g} [k_{d,g} (\vec{N} \cdot \vec{L}) + k_{s,g} (\vec{R} \cdot \vec{V})^n] \\ I_b &= I_{a,b} k_{a,b} + I_{p,b} [k_{d,b} (\vec{N} \cdot \vec{L}) + k_{s,b} (\vec{R} \cdot \vec{V})^n] \end{aligned}$$

### 12.5.8 Shading

Il modello di illuminazione di Phong ci spiega come calcolare l'interazione tra la luce e il materiale senza utilizzare l'Equazione di Rendering (troppo complessa). Ora, vedremo dove calcolare l'equazione (per ogni faccia? per ogni vertice?). Problema del **Flat Shading**: La discretizzazione non rappresenta bene la continuità sulla superficie.

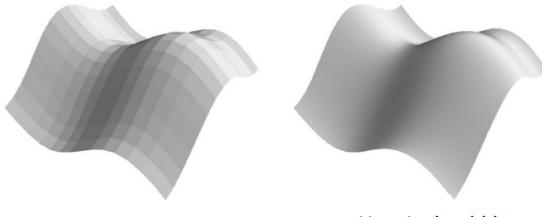


Figure 120: Flat Shading

Soluzione: Userò un'enorme quantità di facce... NON FUNZIONA! Le discontinuità tra le facce sono sempre visibili a causa dell'effetto visivo della Mach Banding.



Figure 121: Mach Banding

## 12.6 Gouraud Shading

I contrasti locali cambiano la nostra percezione. Un oggetto vicino a un oggetto luminoso appare più scuro, un oggetto vicino a un oggetto scuro appare più chiaro. Calcolare l'illuminazione solo in alcuni punti Interpolazione lineare L'interpolazione può essere calcolata durante la rasterizzazione in modo semplice ed efficiente. Lo shading può essere aggiornato incrementalmente durante la rasterizzazione.

### Vertex Normal:

- La normale della faccia è ben definita.
- Un modo è calcolare la normale del vertice come la media delle normali delle facce incidenti sul vertice.

$$\vec{N}_v = \frac{\sum_i \vec{N}_i}{|\sum_i \vec{N}_i|}$$

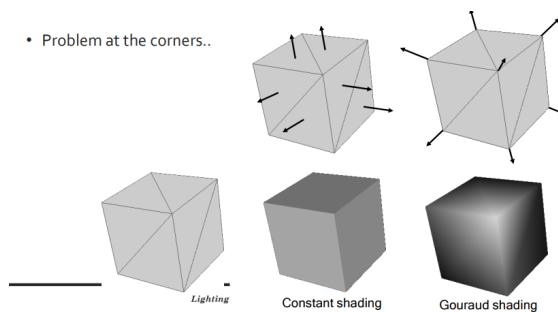


Figure 122: Limitazione

Si può utilizzare il Gouraud Shading, Soluzione: utilizzare normali diverse per le diverse facce degli angoli. La struttura dati dovrebbe tenerne conto.

*Flat vs Gouraud*

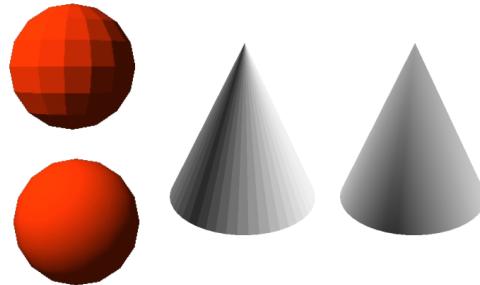


Figure 123: Flat vs Gouraud

#### 12.6.1 Phong Shading

Shading di Gouraud: ottimo rapporto tra complessità e qualità del risultato finale.

- Il risultato finale non è buono per superfici con un alto coefficiente speculare.
- Problema: con un alto indice di riflessione ( $n$ ), le dimensioni dell'illuminazione speculare sono ridotte; utilizzando questa tecnica di shading, l'illuminazione speculare potrebbe propagarsi sull'intera faccia (a causa dell'interpolazione). L'illuminazione speculare non viene disegnata se è all'interno di una faccia.

**Soluzione:** le normali delle superfici vengono interpolate e l'equazione di illuminazione viene calcolata per ogni pixel.

## 13 Scientific Visualization

Le discipline che studiano le tecniche informatiche per la generazione di rappresentazioni visive interattive di dati spazio-temporali acquisiti o simulati, con un collegamento naturale al mondo tridimensionale, sono conosciute come "visualizzazione scientifica". Questa attività umana ha una storia che precede di migliaia di anni la visualizzazione come disciplina nell'ambito dell'informatica. L'illustrazione e la comunicazione visiva della conoscenza fanno parte della nostra storia. I dati di dimensioni sempre crescenti rendono necessario un approccio grafico. Il moderno processo di visualizzazione consiste nel prendere dati grezzi (acquisiti tramite sensori o simulati) e convertirli in una forma comprensibile per gli esseri umani.

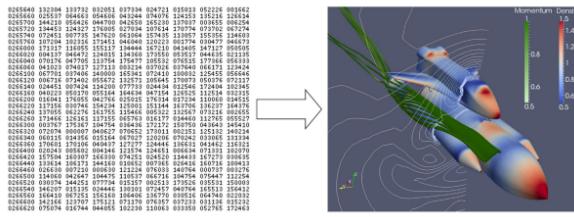


Figure 124: Scientific Visualization

**1987:** La National Science Foundation degli Stati Uniti avvia "Visualizzazione nella computazione scientifica" come nuova disciplina, e un panel dell'ACM conia il termine "visualizzazione scientifica".

- L'interesse è stato stimolato dall'aumento della potenza delle workstation degli scienziati, dai progressi negli algoritmi e dall'aumento delle dimensioni dei set di dati.
- La visualizzazione scientifica è definita come l'uso della grafica computerizzata per l'analisi e la presentazione di dati scientifici calcolati o misurati (Oxford English Dictionary, 1989).
- Nel 1990 si sono tenute le prime conferenze dedicate.
- Oggi è diventata una disciplina matura a sé stante.

Applicazioni nell'Ingegneria, Medicina, Scienza Esempio: La Tomografia Computerizzata (TC) a raggi X e la Risonanza Magnetica (RM) generano immagini (sezioni trasversali del paziente) che vengono combinate per produrre una rappresentazione volumetrica: da numeri a valori in scala di grigi. Applicazioni nell'Ingegneria, Medicina, Scienza Esempio: Dinamica dei fluidi computazionale (flusso d'aria, temperatura mappatata a colori). La Visualizzazione Scientifica riguarda anche la definizione di algoritmi efficienti per la manipolazione interattiva e l'esplorazione dei dati e delle loro caratteristiche.

## 13.1 Basi di Scientific Visualization

Nella Visualizzazione Scientifica, un dataset è dato da un oggetto geometrico di input (il dominio) su cui è definita una funzione, rappresentante gli attributi che desideriamo analizzare e visualizzare. Le tecniche di visualizzazione dipendono dal tipo di dominio e dalla funzione.

$$f : D \rightarrow A$$

$D$  è il dominio,  $A$  è il dominio e  $f$  è la funzione(attributi).

### 13.1.1 Domain

Dimensione (superficie 2D, volumi 3D), Discretizzazione, Geometria (incorporamento) I dati sono rappresentati da un insieme finito di campioni. I dati sono estesi nello spazio tramite schemi di interpolazione. I dati hanno informazioni sulla posizione (incorporamento) e informazioni sulla connettività. Diverse discretizzazioni, in base all'incorporamento e alla connettività Domini strutturati vs non strutturati, connettività fissa vs arbitraria.

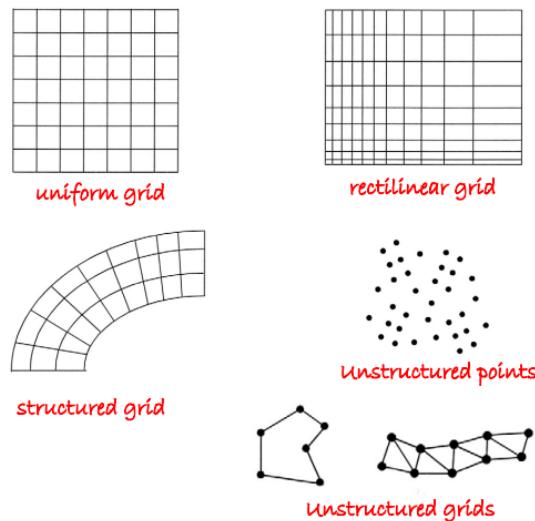


Figure 125: Domain discretization

- **Griglie regolari uniformi:** anche chiamate "raster" (immagine a griglia 2D, volume a griglia 3D): tessellazioni del piano/spazio euclideo mediante elementi quadrati/cubici. I punti sono regolarmente spaziati lungo ciascuna direzione i-j-k, sistema di coordinate parallelo al sistema di coordinate globale x-y-z. Attributi collegati agli elementi della griglia. Rappresentazione semplice, ma "maledizione della dimensionalità".
- **Griglie rettilinee:** tessellazioni mediante rettangoli/cuboidi rettangolari (non necessariamente congruenti tra loro); griglie regolari, ma lo spaziamento dei punti lungo gli assi può variare. Utile per adattare il campionamento alla geometria dei dati. La connettività rimane fissa, ma è necessario conservare informazioni sullo spaziamento.

- **Griglia curvilinea:** stessa connettività delle griglie uniformi/rettilinee, ma geometria irregolare (i punti possono essere posizionati a coordinate arbitrarie - senza sovrapposizioni e auto-intersezioni).
- **Unstructured Points:** Nessuna informazione esplicita sulla connettività, geometria irregolare (chiamata anche nuvole di punti).
- **Unstructured Grids:** connettività e geometria irregolare, combinazioni differenti di celle permettono diverse scelte: triangoli e forme tetraedriche

### 13.1.2 Attributi

Dimensione co-dominio (scalari, vettori, tensori) Statico vs dipendente dal tipo (si applica solo agli attributi o sia al dominio che agli attributi) Deterministico vs incerto (ad esempio, a causa del campionamento dello spazio dei parametri nelle simulazioni - osservazioni dell'insieme). La visualizzazione di dati incerti è un'area di ricerca a sé stante. Gli attributi sono rappresentati da una funzione scalare.

### 13.1.3 Transfer functions

Come codificare i valori scalari? Idea intuitiva: utilizzare una mappa dei colori attraverso un canale percettivo. Disporre in modo continuo una tavolozza di colori e mapparla sulla retta reale.

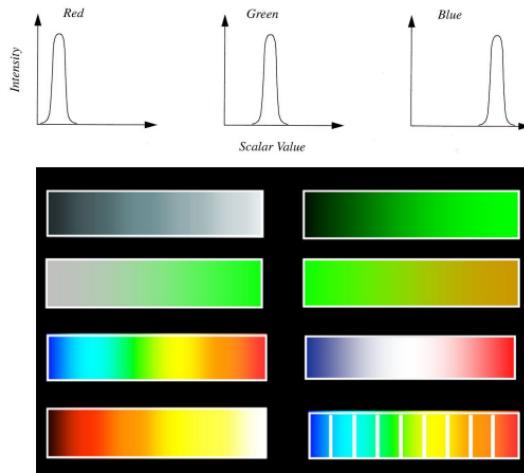


Figure 126: Transfer functions

Per i volumi, le funzioni di trasferimento possono mappare gli scalari al colore e alla trasparenza. Le funzioni di trasferimento sono difficili da progettare. Esistono strumenti semi-automatici (gallerie di progettazione delle funzioni di trasferimento). Il conoscere il dominio di applicazione gioca un ruolo. L'assegnazione di colore e trasparenza alla densità è anche conosciuta come classificazione.

### 13.1.4 Volume Rendering

Molte tecniche. L'idea alla base del "volume ray casting": ogni punto/cella dovrebbe contribuire; combinare le contribuzioni mediante integrazione.

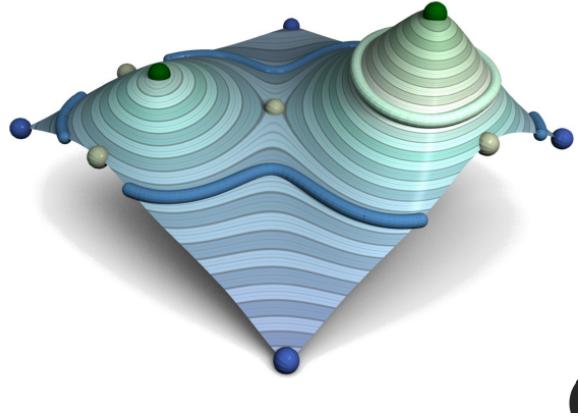


Figure 127: Scalar Field Visualization

Visualizzazione dei punti critici, delle isoline e delle isosuperfici. Visualizzazione delle astrazioni topologiche

### 13.1.5 Critical Points, continuous Case

Nel contesto continuo, i punti critici sono punti di una varietà dove il gradiente di una funzione scalare continua si annulla.

Data Una funzione smmoth  $f : M \rightarrow R$  Su una varietà liscia M, un punto P

- is a critical point if

$$\frac{\partial f}{\partial x_1}(p) = 0, \frac{\partial f}{\partial x_2}(p) = 0, \dots, \frac{\partial f}{\partial x_k}(p) = 0$$

- and a non degenerate critical point if

$$|H_f(p)| = \left| \frac{\partial^2 f}{\partial x_i \partial x_j}(p) \right| \neq 0$$

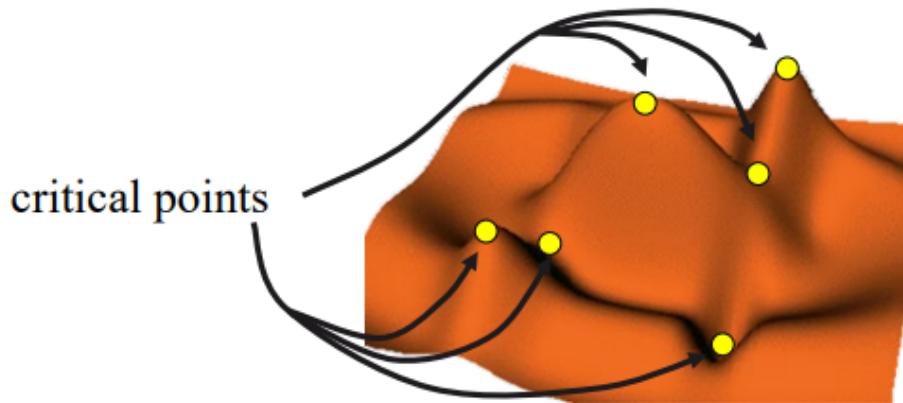


Figure 128: Critical Points

### 13.1.6 Morse functions

Una funzione liscia è chiamata Morse se tutti i suoi punti critici sono non degeneri.

- Proprietà interessanti:
  - Le funzioni di Morse sono un sottoinsieme denso di tutte le funzioni lisce.
  - I punti critici sono isolati.
  - Relazione tra i punti critici e la topologia della varietà.

### 13.1.7 Pice-wise linera (PL) settings

Campo scalare PL: valori della funzione forniti sui vertici e interpolati ovunque altro utilizzando le coordinate baricentriche. La nozione di punto critico si traduce direttamente dal contesto liscio a quello PL? Svantaggio: il gradiente di un campo scalare PL è costante per pezzi (cioè costante all'interno di ciascun simplesso). Abbiamo bisogno di una definizione alternativa per i punti critici. Utilizziamo la connettività del link inferiore e superiore.

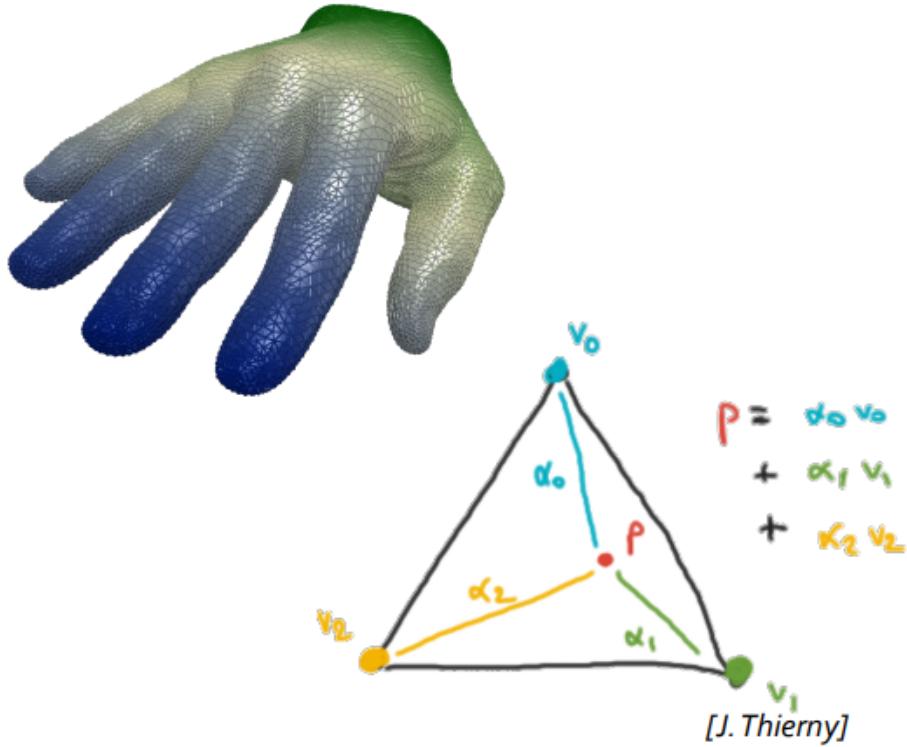


Figure 129: Piece-wise liner setting

Il lower link  $LK_-(v)$  (rispettivamente, l'upper link  $LK_+(v)$ ) di un vertice  $v$  relativo a un campo scalare  $PL f$  è il sottoinsieme del link di  $v$  tale che ciascuna delle sue 0-sottofacce abbia un valore  $f$  strettamente inferiore (rispettivamente, superiore) a  $v$ .

I vertici possono essere classificati come regolari o critici in base alla connettività del link superiore e inferiore. Un vertice  $v$  è regolare se sia il lower link  $LK_-(v)$  che l'upper link  $LK_+(v)$  sono semplicemente connessi. In caso contrario,  $v$  è un vertice critico e  $f(v)$  è un valore critico iso (in opposizione a un valore iso regolare). Campo scalare  $PL$  Valori critici distinti Nessun punto critico degenerato (in pratica: qualsiasi campo scalare  $PL$  dopo una perturbazione) Problema: in pratica, i punti critici possono comparire in corrispondenza di lievi onde della funzione dovute al rumore nel processo di generazione dei dati (rumore di acquisizione, rumore numerico nella simulazione).

- Per rendere i punti critici affidabili e utili nella pratica, è necessario un meccanismo per classificare i punti critici come rumore o segnale.
- Possiamo utilizzare l'Omelogia Persistente.

### 13.1.8 Filtration

Intuizione: sequenze annidate di insiemi di sotto livelli, in accordo all'aumento del valore scalare  $PL.Q$ . Come cambia il valore della funzione, gli attributi topologici appaiano si uniscono o scompaiono, questi cambiamenti avvengono nei punti critici. Dato che questi eventi occorrono nei punti critici, possiamo associare una misura di persistenza con le

coppie di punti critici dell'input degli attributi scalari. **persistenza**: valore assoluto della differenza di f-value di coppie di punti critici. **Intuizione**: lifespan.



Figure 130: Pice-wise liner setting

### 13.1.9 Diagramma di Persistenza

La persistenza è una misura dell'*importanza* di un punto critico, ogni coppia di punti critici è rappresentata da una barra verticale e la persistenza è data dall'altezza della barra. Trovare un attributo scalare con solo un sottoinsieme di punti critici, la persistenza è il criterio. Il diagramma persistenza mostra una rappresentazione concisa di coppie di punti critici, ma non mostrano informazioni sulla relazione di adiacenza di coppie di punti del dominio ???.

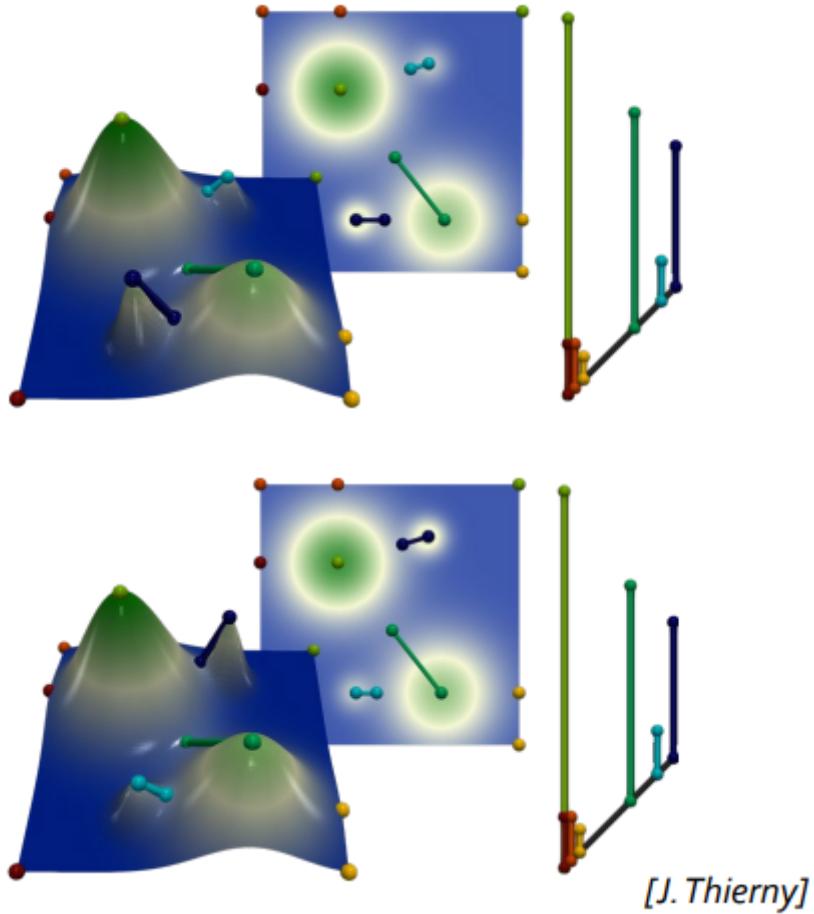


Figure 131: Diagramma della persistenza

Idea: guardare ai level set cioè l'insieme di punti che hanno lo stesso valore della funzionem, un atrlo strumento sono i Reeb graphs, che tengono traccia dell'evoluzione dei level sets.

### 13.1.10 Reeb graph

In breve, il grafo di Reeb contrae continuamente ciascun componente连nesso dei level set in un punto, producendo un complesso simpliciale di dimensione 1. Di nuovo, la topologia dei level set cambia nei punti critici. Gli estremi vengono mappati su vertici di valenza -1, mentre le sella vengono mappate su vertici di valenza superiore.



[J. Thierry]

Figure 132: Reeb Graph

Una gerarchia di Reeb Graphs può essere ottenuta dalla rimozione ripetuta, guidata dalla persistenza, di coppie di punti critici( quindi archi nel grafo). La segmentazione può essere ottenuta considerando la pre immagine di ogni arco nel Reeb graph.

### 13.2 Vector field visualization

Gli attributi sono rappresentati da vettori:

- rateo di cambiamento ...
- i vettori hanno grandezza e direzione.

Con problemi che coinvolgono le derivate dei campi scalari. Il modo più semplice: disegnare frecce Nessuna interpolazione richiesta, impressione generale del campo, ma problemi con le occlusioni La grandezza mappata a lunghezza, larghezza, colore 3D: anche distorsione prospettica (interazione necessaria per recuperare indizi di profondità)

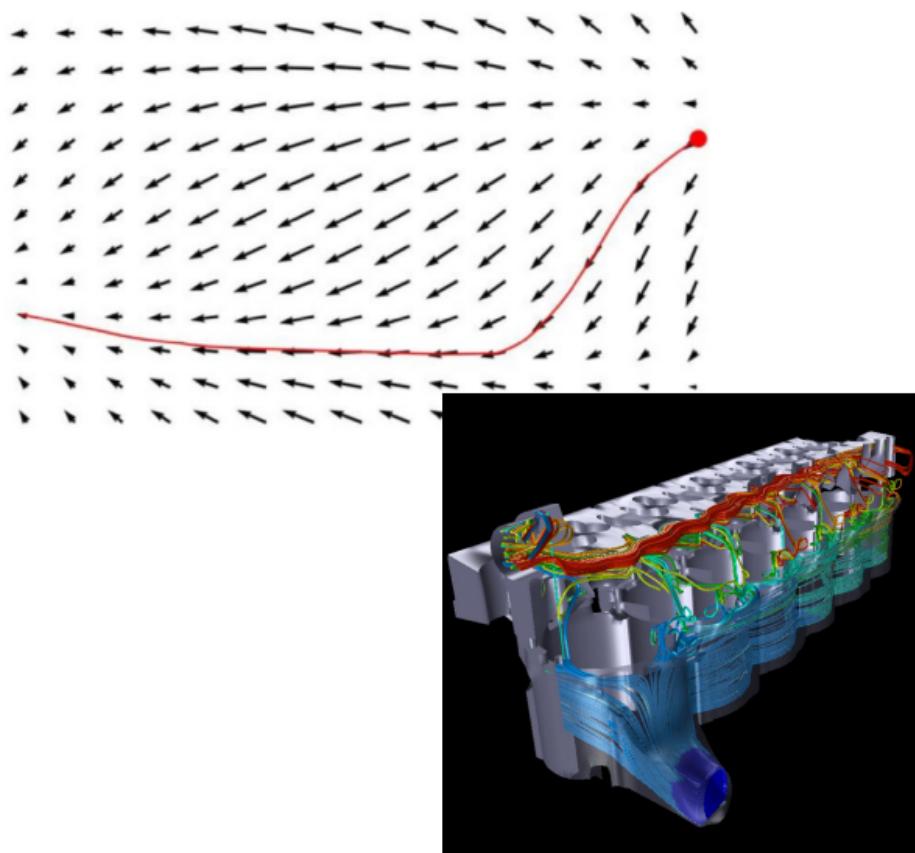


Figure 133: Vector field visualization

### 13.2.1 Tensor and Vector field visualization

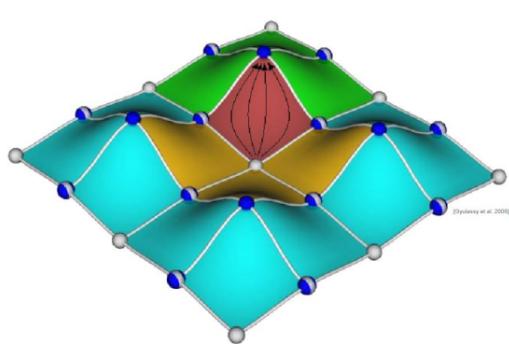


Figure 134: Topological Data Analysis

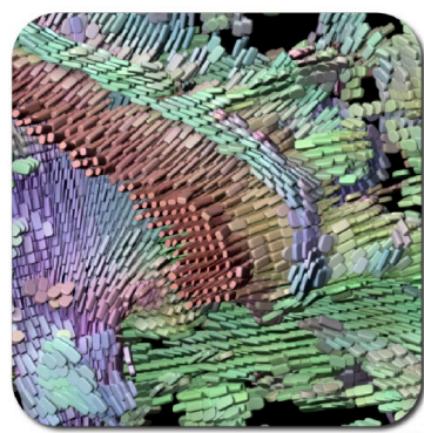
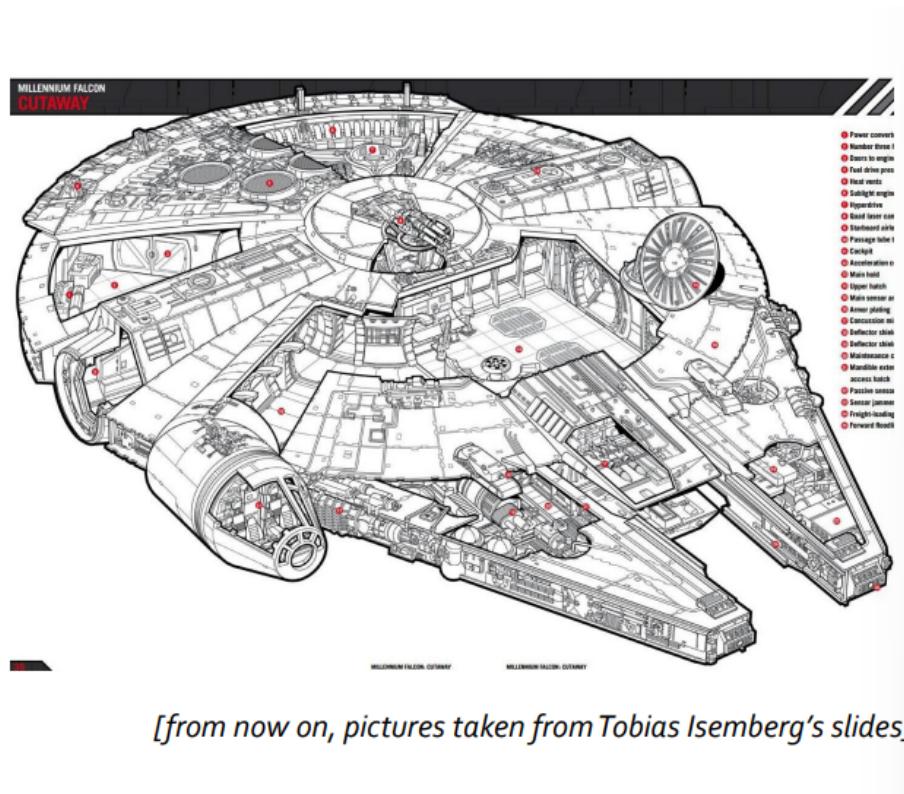


Figure 135: Attributi rappresentati da tensori con problemi che coinvolgono derivate dei campi vettoriali

### 13.3 Scientific illustration

La scientific illustration è una disciplina che si concentra sul realismo, la scienza e l'arte: utilizzando disegni o rendering di immagini di soggetti scientifici per informare e comunicare. Artwork e scienze: skills estetiche in combinazione con osservazioni scientifiche, ne sono esempi illustrazioni biologiche, mediche e tecniche.



*[from now on, pictures taken from Tobias Isenberg's slides]*

Figure 136: illustration visualization

In contrasto con la computer grafica tradizionale che si concentra sul realismo, la **NPR**(Non-photorealistic rendering) è ispirata dalla pittura, dal disegno e dai cartoni. Esempi tecnici per l'NPR includono:

- **Sparse line drawings:** disegni di insiemi sparsi di contorni lineari, che illustrano alcune feature.
- **Stippling:** creazione di pattern unici usando punti, che possono simulare vari livelli di ombra e solidità.

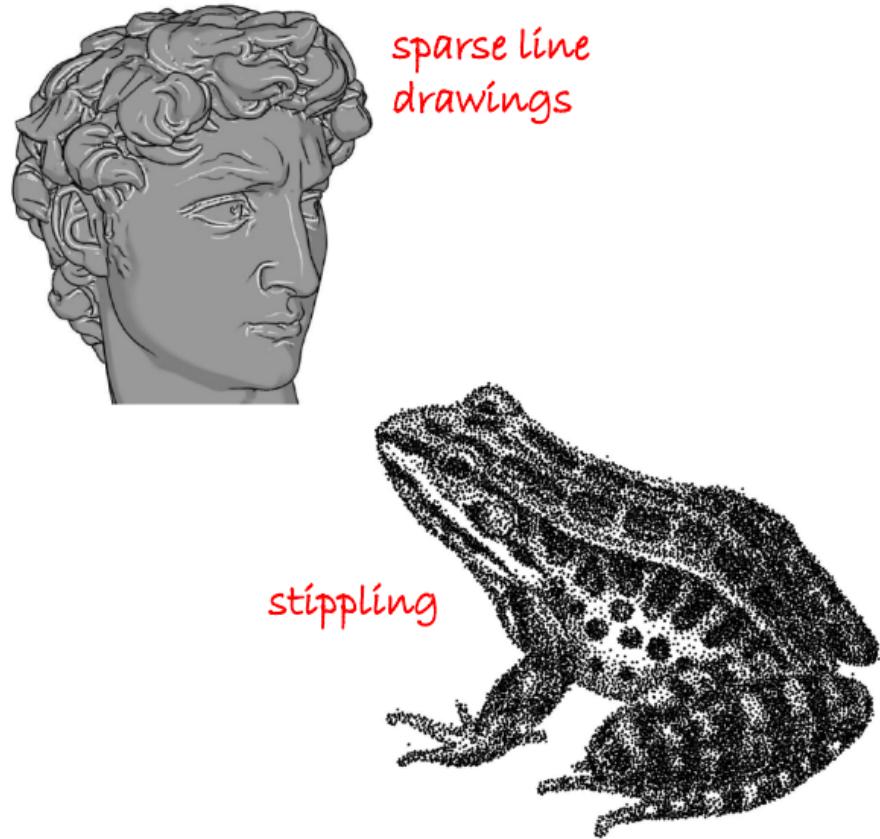


Figure 137: Example: Non-photorealistic rendering

Si concentra su diversi livelli di oggetti o livelli.

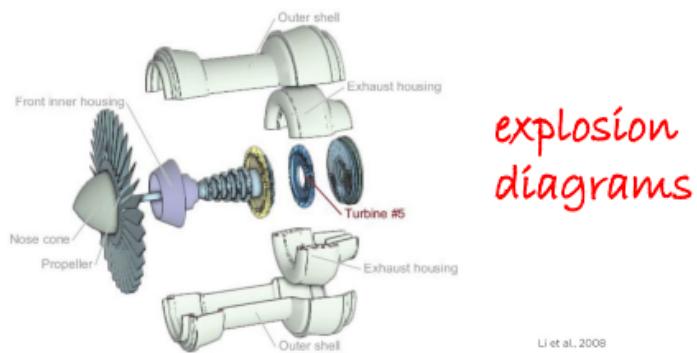
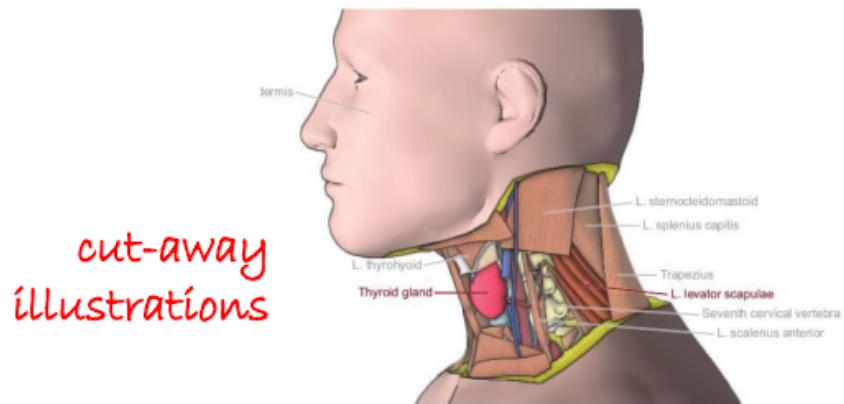


Figure 138: Expressive Viusalization