Template Matching Techniques in Computer Vision

Roberto Brunelli

FBK - Fondazione Bruno Kessler

1 Settembre 2008

Table of contents

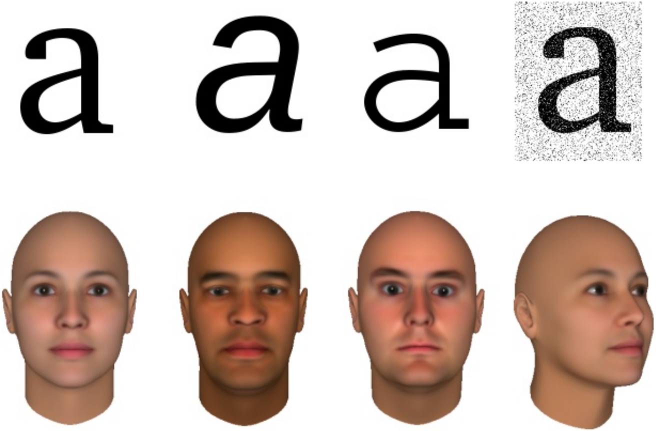
Template matching

template/pattern

1. anything fashioned, shaped, or designed to serve as a model from which something is to be made: a model, design, plan, outline;
2. something formed after a model or prototype, a copy; a likeness, a similitude;
3. an example, an instance; esp. a typical model or a representative instance;

matching to compare in respect of similarity; to examine the likeness of difference of.

... template variability ...



... and Computer Vision

Many important computer vision tasks can be solved with template matching techniques:

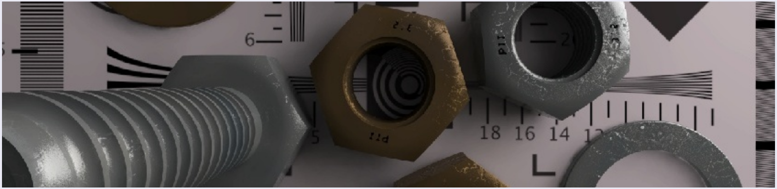
- Object detection/recognition
- Object comparison
- Depth computation

and template matching depends on

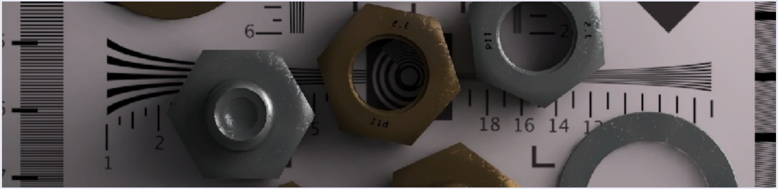
- Physics (imaging)
- Probability and statistics
- Signal processing

Imaging

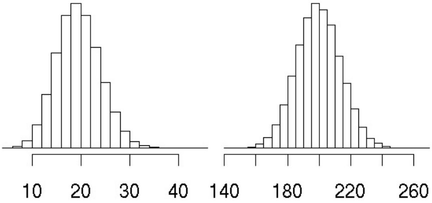
Perspective camera



Telecentric camera



Imaging



Photon noise (Poisson)

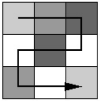
Quantum nature of light results in appreciable photon noise[a]

$$p(n) = e^{-(r\Delta t)} \frac{(r\Delta t)^n}{n!}$$

$$\text{SNR} \leq \frac{I}{\sigma_I} = \frac{n}{\sqrt{n}} = \sqrt{n}$$

[a] r photons per unit time, Δt gathering time

Finding them ...

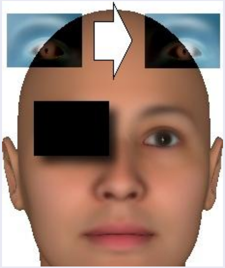


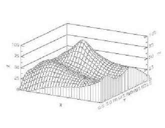
60 40 20 100 20 ...

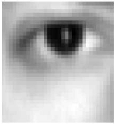
$$d(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

$$s(\mathbf{x},\mathbf{y}) = \frac{1}{1 + d(\mathbf{x},\mathbf{y})}$$

A sliding window approach



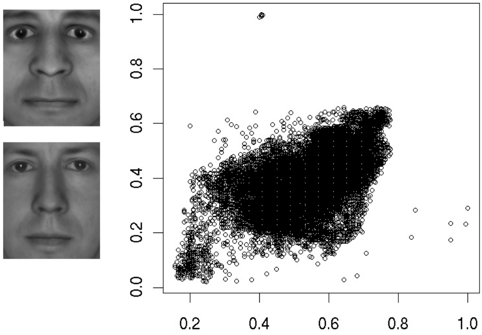




... robustly

Specularities and noise can result in outliers: abnormally large differences that may adversely affect the comparison.

Specularities outliers



... robustly

We downweight outliers changing the metrics:

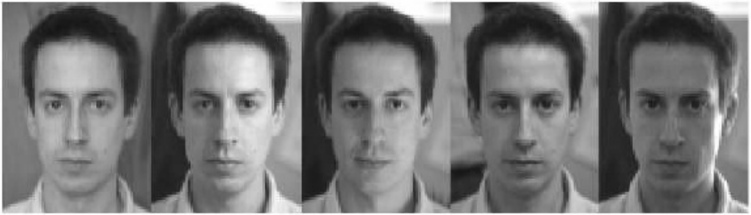
$$\sum_{i=1}^{N}(z_i)^2 \rightarrow \sum_{i=1}^{N}\rho(z_i), \quad z_i = x_i - y_i$$

with one that has a more favourable influence function

$$\psi(z) = \frac{d\rho(z)}{dz}$$

$$\begin{aligned} \rho(z) &= z^2 & \psi(z) &= z \\ \rho(z) &= |z| & \psi(z) &= \mathrm{sign} z \\ \rho(z) &= \log\left(1 + \frac{z^2}{a^2}\right) & \psi(z) &= \frac{z}{a^2 + z^2} \end{aligned}$$

Illumination effects



Additional Template variability

Illumination variations affect images in a complex way, reducing the effectiveness of template matching techniques

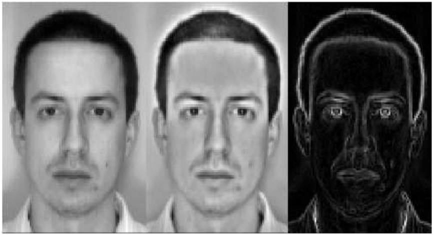
Contrast and edge maps

Image transforms such as local contrast can reduce the effect of illumination:

$$
\begin{aligned}
N' &= \frac{I}{I * K_\sigma} \\
N &= \begin{cases} N' & \text{if } N' \leq 1 \\ 2 - \frac{1}{N'} & \text{if } N' > 1 \end{cases}
\end{aligned}
$$

$$(f * g)(x) = \int f(y) g(x - y)\, dy$$

Local contrast and edge maps

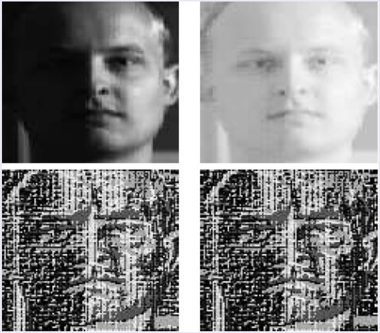


Ordinal Transforms

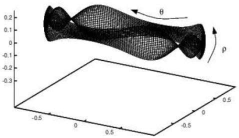
Let us consider a pixel $I(\boldsymbol{x})$ and its neighborhood of $W(\boldsymbol{x}, l)$ of size l. Denoting with $\otimes$ the operation of concatenation, the Census transform is defined as

$$C(\boldsymbol{x}) = \bigotimes_{\boldsymbol{x}' \in W(\boldsymbol{x}, l) \setminus \boldsymbol{x}} \theta(I(\boldsymbol{x}) - I(\boldsymbol{x}'))$$

CT invariance



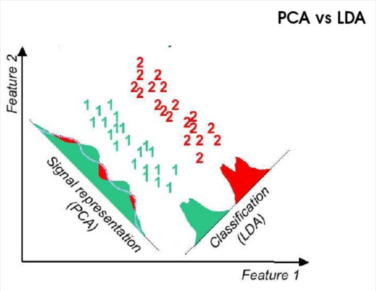
Matching variable patterns



[a]

Patterns of a single class may span a complex manifold of a high dimensional space: we may try to find a compact space enclosing it, possibly attempting multiple local linear descriptions.

Different criteria, different basis



[a] step edge, orientation θ and axial distance ρ

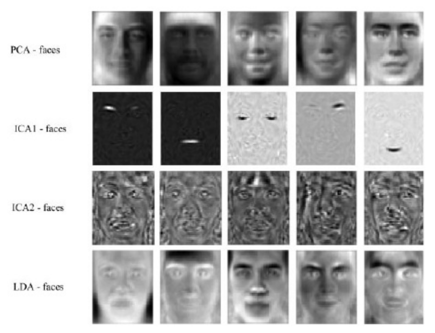
Subspaces approaches

PCA the eigenvectors of the covariance matrix;

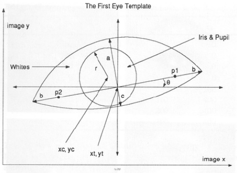
ICA the directions onto which data projects with maximal non Gaussianity;

LDA the directions maximizing between class scatter over within class scatter.

PCA, ICA (I and II), LDA



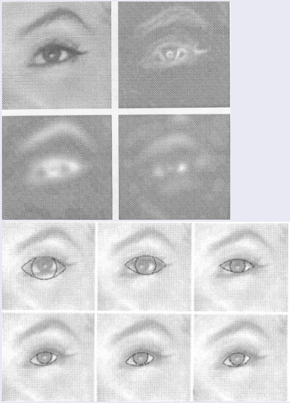
Deformable templates



1. The circle representing the iris, characterized by its radius r and its center $\mathbf{x}_c$. The interior of the circle is attracted to the low intensity values while its boundary is attracted to edges in image intensity.

$$k_v = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Eyes potentials



Deformable templates

Diffeomorphic matching[a]:

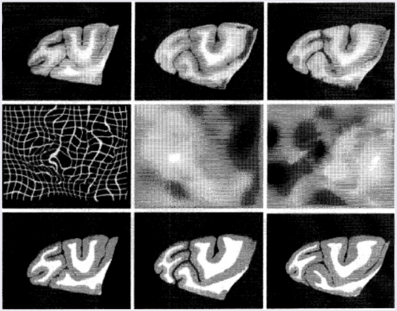
$$A \circ \boldsymbol{u}(\boldsymbol{x}) = A(u(\boldsymbol{x})) \approx B(\boldsymbol{x})$$

$$\hat{\boldsymbol{u}} = \operatorname*{argmin}_{\boldsymbol{u}} \int_\Omega \Delta(A \circ \boldsymbol{u}, B; \boldsymbol{x}) d\boldsymbol{x} + \Delta(\boldsymbol{u})$$

$$\begin{aligned} \Delta(A, B) &= \int_\Omega (A(\boldsymbol{x}) - B(\boldsymbol{x}))^2 \, d\boldsymbol{x} \\ \Delta(\boldsymbol{u}) &= \|\boldsymbol{u} - \boldsymbol{I}_{\boldsymbol{u}}\|_\Omega^{H_1} \end{aligned}$$

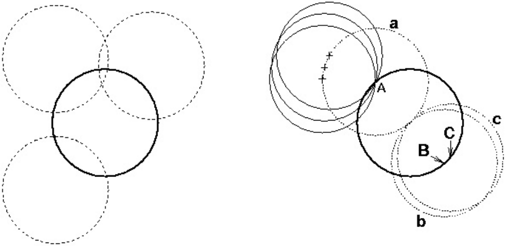
$$\|\boldsymbol{a}\|_\Omega^{H_1} = \int_{\boldsymbol{x}\in\Omega} \|\boldsymbol{a}(\boldsymbol{x})\|^2 + \|\partial(\boldsymbol{u})/\partial(\boldsymbol{x})\|_F^2 d\boldsymbol{x}$$

Brain warping



[a] a bijective map $\boldsymbol{u}(\boldsymbol{x})$ such that both it and its inverse $\boldsymbol{u}^{-1}$ are differentiable

Linear structures: Radon/Hough Transforms



Radon

Hough

$$\mathcal{R}_{s(\boldsymbol{q})}(I;\boldsymbol{q}) = \int_{\mathbb{R}^d} \delta(\mathcal{K}(\boldsymbol{x};\boldsymbol{q}))I(\boldsymbol{x})\ d\boldsymbol{x}$$

In the Radon approach (left), the supporting evidence for a shape with parameter $\boldsymbol{q}$ is collected by integrating over $s(\boldsymbol{q})$. In the Hough approach (right), each potentially supporting pixel (e.g. edge pixels $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$) votes for all shapes to which it can potentially belong (all circles whose centers lay respectively on circles $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$).

Detection as Learning

Given a set $\{(\mathbf{x}_i, y_i)\}_i$, we search a function $\hat{f}$ minimizing the empirical (approximation) squared error

$$\begin{aligned} E^{\mathrm{MSE}}_{\mathrm{emp}} &= \frac{1}{N}\sum_i (y_i - f(\mathbf{x}_i))^2 \\ \hat{f}(\mathbf{x}) &= \operatorname*{argmin}_f E^{\mathrm{MSE}}_{\mathrm{emp}}(f; \{(\mathbf{x}_i, y_i)\}_i) \end{aligned}$$

This ill posed problem can be regularized, turning the optimization problem of Equation 1 into

$$\hat{f}(\lambda) = \operatorname*{argmin}_{f \in \mathfrak{H}} \frac{1}{N}\sum_i (y_i - f(\mathbf{x}_i))^2 + \lambda \|f\|_{\mathfrak{H}}$$

where $\|f\|_{\mathfrak{H}}$ is the norm of f in the (function) space $\mathfrak{H}$ to which we restrict our quest for a solution.

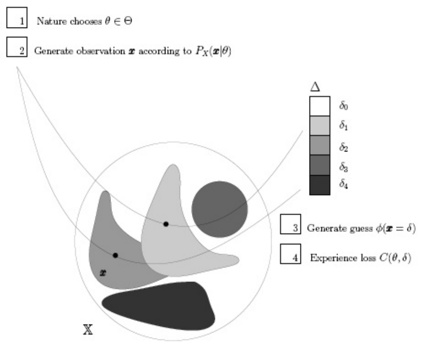
Detection as testing

The problem of template detection fits within game theory.

The game proceeds along the following steps:

1. nature chooses a state $\theta \in \Theta$;
2. a hint x is generated according to the conditional distribution $P_X(x|\theta)$;
3. the computational agent makes its guess $\phi(x) = \delta$;
4. the agent experiences a loss $C(\theta, \delta)$.

Gaming with nature


1 Nature chooses $\theta \in \Theta$
2 Generate observation $\boldsymbol{x}$ according to $P_X(\boldsymbol{x}|\theta)$
Δ
δ_0
δ_1
δ_2
δ_3
δ_4
3 Generate guess $\phi(\boldsymbol{x} = \delta)$
4 Experience loss $C(\theta, \delta)$
$\boldsymbol{x}$
$\mathbb{X}$

Hypothesis testing and Templates

Two cases are relevant to the problem of template matching:

1. $\Delta = \{\delta_0, \delta_1, \ldots, \delta_{K-1}\}$, that corresponds to hypothesis testing, and in particular the case $K = 2$, corresponding to binary hypothesis testing. Many problems of pattern recognition fall within this category.
2. $\Delta = \mathbb{R}^n$, corresponding to the problem of point estimation of a real parameter vector: a typical problem being that of model parameter estimation.

Template detection can be formalized as a binary hypothesis test:

$$\begin{aligned} H_0 : \quad \mathbf{x} \quad &\sim p_\theta(\mathbf{x}), \theta \in \Theta_0 \\ H_1 : \quad \mathbf{x} \quad &\sim p_\theta(\mathbf{x}), \theta \in \Theta_1 \end{aligned}$$

Signal vs. Noise

Template detection in the presence of additive white Gaussian noise $\boldsymbol{\eta} \sim N(\mathbf{0}, \sigma^2 I)$

$$\begin{aligned} H_0 : \quad \boldsymbol{x} &= \boldsymbol{\eta} \\ H_1 : \quad \boldsymbol{x} &= \begin{cases} \boldsymbol{f} + \boldsymbol{\eta} & \text{simple} \\ \alpha \boldsymbol{f} + \boldsymbol{o} + \boldsymbol{\eta} & \text{composite} \end{cases} \end{aligned}$$

An hypothesis test (or classifier) is a mapping ϕ

$$\phi : (\mathbb{R}^{n_d})^N \to \{0, \ldots, M-1\}.$$

The test ϕ returns an hypothesis for every possible input, partitioning the input space into a disjoint collection $R_0, \ldots, R_{M-1}$ of decision regions:

$$R_k = \{(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) | \phi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = k\}.$$

Error types

The probability of a type I (false alarm) P_F (size or α)

$$\alpha = P_F = P(\phi = 1 | H_0)$$

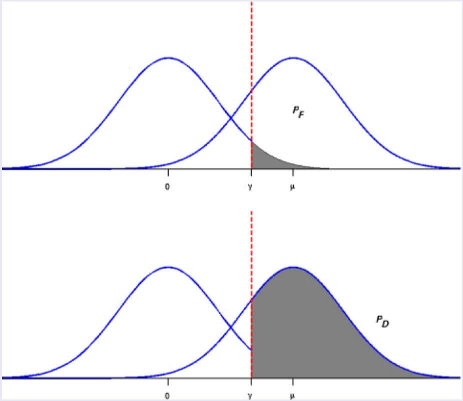
The detection probability P_D (power or β):

$$\beta(\theta) = P_D = P(\phi = 1 | \theta \in \Theta_1),$$

The probability of a type II error, or miss probability P_M is

$$P_M = 1 - P_D.$$

False alarms and detection



The Bayes Risk

The Bayes approach is characterized by the assumption that the occurrence probability of each hypothesis π_i is known a priori.

The optimal test is the one that minimizes the Bayes risk C_B:

$$\begin{aligned} C_B &= \sum_{i,j} C_{ij} P(\phi(\boldsymbol{X}) = i | H_j)\pi_j \\ &= \sum_{i,j} C_{ij} \left(\int_{R_i} p_j(\boldsymbol{x}) d\boldsymbol{x} \right) \pi_j \\ &= \int_{R_0} (C_{00}\pi_0 p_0(\boldsymbol{x}) + C_{01}\pi_1 p_1(\boldsymbol{x}))\, d\boldsymbol{x} + \\ &\quad \int_{R_1} (C_{10}\pi_0 p_0(\boldsymbol{x}) + C_{11}\pi_1 p_1(\boldsymbol{x}))\, d\boldsymbol{x}. \end{aligned}$$

The likelihood ratio

We may minimize the Bayes risk assigning each possible $\boldsymbol{x}$ to the region whose integrand at $\boldsymbol{x}$ is smaller:

$$L(\boldsymbol{x}) \equiv \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\pi_0(C_{10} - C_{00})}{\pi_1(C_{01} - C_{11})} \equiv \nu$$

where $L(\boldsymbol{x})$ is called the likelihood ratio.
When $C_{00} = C_{11} = 0$ and $C_{10} = C_{01} = 1$

$$L(\boldsymbol{x}) \equiv \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\pi_0}{\pi_1} \equiv \nu$$

equivalent to the maximum a posteriori (MAP) rule

$$\phi(\boldsymbol{x}) = \underset{i \in \{0,1\}}{\operatorname{argmax}} \, \pi_i p_i(\boldsymbol{x})$$

Frequentist testing

The alternative to Bayesian hypothesis testing is based on the Neyman-Pearson criterion and follows a classic, frequentist approach based on

$$\begin{aligned} P_F &= \int_{R_1} p_0(\mathbf{x}) d\mathbf{x} \\ P_D &= \int_{R_1} p_1(\mathbf{x}) d\mathbf{x}. \end{aligned}$$

we should design the decision rule in order to maximize P_D without exceeding a predefined bound on P_F:

$$\hat{R}_1 = \operatorname*{argmax}_{R_1 : P_F \leq \alpha} P_D.$$

... likelihood ratio again

The problem can be solved with the method of Lagrange multipliers:

$$\begin{aligned} E &= P_D + \lambda(P_F - \alpha') \\ &= \int_{R_1} p_1(\mathbf{x}) d\mathbf{x} + \lambda \left(\int_{R_1} p_0(\mathbf{x}) d\mathbf{x} - \alpha' \right) \\ &= -\lambda\alpha' + \int_{R_1} \left(p_1(\mathbf{x}) + \lambda p_0(\mathbf{x}) \right) d\mathbf{x} \end{aligned}$$

where $\alpha' \leq \alpha$. In order to maximize E, the integrand should be positive leading to the following condition:

$$\frac{p_1(\mathbf{x})}{p_0(\mathbf{x})} \overset{H_1}{>} -\lambda$$

as we are considering region R_1.

The Neyman Pearson Lemma

In the binary hypothesis testing problem, if $\alpha_0 \in [0, 1)$ is the size constraint, the most powerful test of size $\alpha \leq \alpha_0$ is given by the decision rule

$$\phi(\boldsymbol{x}) = \begin{cases} 1 & \text{if } L(\boldsymbol{x}) > \nu \\ \gamma & \text{if } L(\boldsymbol{x}) = \nu \\ 0 & \text{if } L(\boldsymbol{x}) < \nu \end{cases}$$

where ν is the largest constant for which

$$P_0 \left(L(\boldsymbol{x}) \geq \nu \right) \geq \alpha_0 \text{ and } P_0 \left(L(\boldsymbol{x}) \leq \nu \right) \geq 1 - \alpha_0$$

The test is unique up to sets of probability zero under H_0 and H_1.

An important example

Discriminate two deterministic multidimensional signals corrupted by zero average Gaussian noise:

$$\begin{aligned} H_0 : \quad \boldsymbol{x} \quad &\sim N(\boldsymbol{\mu}_0, \Sigma), \\ H_1 : \quad \boldsymbol{x} \quad &\sim N(\boldsymbol{\mu}_1, \Sigma), \end{aligned}$$

Using the Mahalanobis distance

$$d^2_\Sigma(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{y})$$

we get

$$\begin{aligned} p_0(\boldsymbol{x}) \quad &= \quad \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} d^2_\Sigma(\boldsymbol{x}, \boldsymbol{\mu}_0) \right] \\ p_1(\boldsymbol{x}) \quad &= \quad \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} d^2_\Sigma(\boldsymbol{x}, \boldsymbol{\mu}_1) \right] \end{aligned}$$

... with an explicit solution.

The decision based on the log-likelihood ratio is

$$\phi(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{x}_0) \geq \nu_\Lambda \\ 0 & \boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{x}_0) < \nu_\Lambda \end{cases}$$

with

$$\boldsymbol{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0), \quad \boldsymbol{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)$$

and P_F, P_D depend only on the distance of the means of the two classes normalized by the amount of noise, which is a measure of the SNR of the classification problem. When $\Sigma = \sigma^2 I$ and $\boldsymbol{\mu}_0 = \mathbf{0}$ we have matching by projection:

$$r_u = \boldsymbol{\mu}_1^T \boldsymbol{x} \underset{H_0}{\overset{H_1}{\gtrless}} \nu'_\Lambda$$

… more details

$$
\begin{aligned}
P_F &= P_0(\Lambda(\boldsymbol{x}) \geq \nu) = Q\left(\frac{\nu + \sigma_0^2/2}{\sigma_0}\right) = Q(z) \\
P_D &= P_1(\Lambda(\boldsymbol{x}) \geq \nu) = Q\left(\frac{\nu - \sigma_0^2/2}{\sigma_0}\right) = Q(z - \sigma_0) \\
\sigma_0^2(\Lambda(\boldsymbol{x})) &= \sigma_1^2(\Lambda(\boldsymbol{x})) = \boldsymbol{w}^T \Sigma \boldsymbol{w} \\
z &= \nu/\sigma_0 + \sigma_0/2
\end{aligned}
$$

Variable patterns ...

A common source of signal variability is its scaling by an unknown gain factor α possibly coupled to a signal offset β

$$\boldsymbol{x}' = \alpha \boldsymbol{x} + \beta \mathbf{1}$$

A practical strategy is to normalize both the reference signal and the pattern to be classified to zero average and unit variance:

$$\begin{aligned}
\boldsymbol{x}' &= \frac{(\boldsymbol{x} - \bar{x})}{\sigma_x} \\
\bar{x} &= \frac{1}{n_d} \sum_{i=1}^{n_d} x_i \\
\sigma_x &= \frac{1}{n_d} \sum_{i=1}^{n_d} (x_i - \bar{x})^2 = \frac{1}{n_d} \sum_{i=1}^{n_d} x_i^2 - \bar{x}^2
\end{aligned}$$

Correlation

or, equivalently, replacing matching by projection with

$$r_{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_i (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_i (x_i - \mu_x)^2} \sqrt{\sum_i (y_i - \mu_y)^2}}$$

which is related to the fraction of the variance in y accounted for by a linear fit of x to y $\hat{\boldsymbol{y}} = \hat{a}\boldsymbol{x} + \hat{b}$

$$r_P^2 = 1 - \frac{s_{y|x}^2}{s_y^2}$$

$$\begin{aligned} s_{y|x}^2 &= \sum_{i=1}^{n_d} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n_d} \left(y_i - \hat{a}x_i - \hat{b} \right)^2 \\ s_y^2 &= \sum_{i=1}^{n_d} (y - \bar{y})^2 \end{aligned}$$

(Maximum likelihood) estimation

The likelihood function is defined as

$$l(\boldsymbol{\theta}|\{\boldsymbol{x}_i\}_{i=1}^N) = \prod_{i=1}^N p(\boldsymbol{x}_i|\boldsymbol{\theta})$$

where $\boldsymbol{x}^N = \{\boldsymbol{x}_i\}_{i=1}^N$ is our (fixed) dataset and it is considered to be a function of $\boldsymbol{\theta}$. The maximum likelihood estimator (MLE) $\hat{\boldsymbol{\theta}}$ is defined as

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} l(\boldsymbol{\theta}|\boldsymbol{x}^N)$$

resulting in the parameter that maximizes the likelihood of our observations.

Bias and Variance

Definition

The bias of an estimator $\hat{\theta}$ is

$$\mathrm{bias}(\hat{\theta}) = E(\hat{\theta}) - \theta$$

where θ represents the true value. If $\mathrm{bias}(\hat{\theta}) = 0$ the operator is said to be unbiased.

Definition

The mean squared error (MSE) of an estimator is

$$\mathrm{MSE}(\hat{\theta}) = E((\hat{\theta} - \theta)^2)$$

MLE properties

1. The MLE is asymptotically unbiased, i.e., its bias tends to zero as the number of samples increases to infinity.
2. The MLE is asymptotically efficient: asymptotically, no unbiased estimator has lower mean squared error than the MLE.
3. The MLE is asymptotically normal.

Shrinkage (James-Stein estimators)

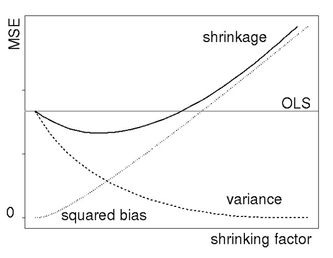
$$\mathrm{MSE}(\hat{\theta}) = \mathrm{var}(\hat{\theta}) + \mathrm{bias}^2(\hat{\theta})$$

We may reduce MSE trading off bias for variance, using a linear combination of estimators T and S

$$T_s = \lambda T + (1-\lambda)S$$

shrinking S towards T.

Shrinkage



James-Stein Theorem

Let $\boldsymbol{X}$ be distributed according to a n_d-variate normal distribution $N(\boldsymbol{\theta}, \sigma^2 I)$. Under the squared loss, the usual estimator $\boldsymbol{\delta}(\boldsymbol{X}) = \boldsymbol{X}$ exhibits a higher loss for any $\boldsymbol{\theta}$, being therefore dominated, than

$$\boldsymbol{\delta}_a(X) = \boldsymbol{\theta}_0 + \left(1 - \frac{a\sigma^2}{\|\boldsymbol{X} - \boldsymbol{\theta}_0\|^2}\right)(\boldsymbol{X} - \boldsymbol{\theta}_0)$$

for $n_d \geq 3$ and $0 < a < 2(n_d - 2)$ and $a = n_d - 2$ gives the uniformly best estimator in the class. The risk of $\delta_{n_d - 2}$ at $\boldsymbol{\theta}_0$ is constant and equal to $2\sigma^2$ (instead of $n_d\sigma^2$ of the usual estimator).

JS estimation of covariance matrices

The unbiased sample estimate of the covariance matrix is

$$\hat{\Sigma} = \frac{1}{N-1} \sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$$

and it benefits from shrinking in the small sample, high dimensionality case, avoiding the singularity problem. The optimal shrinking parameter can be obtained in closed form for many useful shrinking targets.

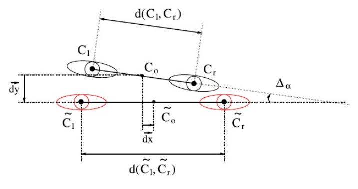
Significant improvements are reported in template (face) detection tasks using similar approaches.

Error breakdown

Detailed error breakdown can be exploited to improve system performance.

Error measures should be invariant to translation, scaling, rotation.

Eyes localization errors

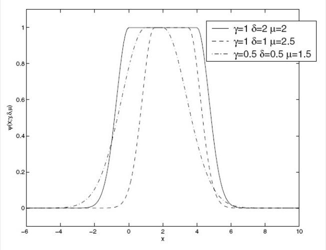


Error scoring

Error weighting or scoring functions can be tuned to tasks: errors are mapped into the range $[0, 1]$, the lower the score, the worse the error.

A single face detection system can be scored differently when considered as a detection or localization system by changing the parameters controlling the weighting functions, using more peaked scoring functions for localization.

Task selective penalties



Error impact

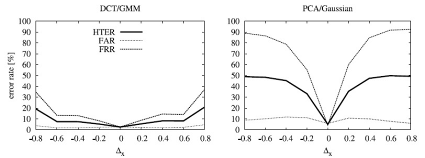
The final verification error Δ_v

$$\Delta_v(\{\boldsymbol{x}_i\}) = \sum_i f(\boldsymbol{\delta}(\boldsymbol{x}_i); \boldsymbol{\theta})$$

must be expressed as a function of the detailed error information that can be associated to each localization $\boldsymbol{x}_i$:

$$(\delta_{x_1}(\boldsymbol{x}_i), \delta_{x_2}(\boldsymbol{x}_i), \delta_s(\boldsymbol{x}_i), \delta_\alpha(\boldsymbol{x}_i)).$$

System impact



δ_{x_1}, face verification systems, FAR=false acceptance/impostors, FRR=false rejections/true client, HTER= (FAR+FRR)/2

Training and testing: concepts

Let $\mathcal{X}$ be the space of possible inputs (without label), $\mathcal{L}$ the set of labels, $\mathcal{S} = \mathcal{X} \times \mathcal{L}$ the space of labeled samples, and $D = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N\}$, where $\boldsymbol{s}_i = (\boldsymbol{x}_i, l_i) \in \mathcal{S}$, be our dataset.

A classifier is a function $\mathfrak{C} : \mathcal{X} \rightarrow \mathcal{L}$, while an inducer is an operator $\mathfrak{I} : D \rightarrow \mathfrak{C}$ that maps a dataset into a classifier.

... and methods

The accuracy ϵ of a classifier is the probability $p(\mathfrak{C}(\boldsymbol{x}) = l, (\boldsymbol{x}, l) \in S)$ that its label attribution is correct. The problem is to find a low bias and low variance estimate $\hat{\epsilon}(\mathfrak{C})$ of ϵ. There are three main different approaches to accuracy estimation and model selection:

1. hold-out,
2. bootstrap,
3. *k*-fold cross validation.

Hold Out

A subset D_h of n_h points is extracted from the complete dataset and used as testing set while the remaining set $D_t = D \setminus D_h$ of $N - n_h$ points is provided to the inducer to train the classifier. The accuracy is estimated as

$$\hat{\epsilon}_h = \frac{1}{n_h} \sum_{\boldsymbol{x}_i \in D_h} \delta[J(D_t; \boldsymbol{x}_i), l_i]$$

where $\delta(i,j) = 1$ when $i = j$ and 0 otherwise. It (approximately) follows a Gaussian distribution $N(\epsilon, \epsilon(1-\epsilon)/n_h)$, from which an estimate of the variance (of ϵ) follows.

Bootstrap

The accuracy and its variance are estimated from the results of the classifier over a sequence of bootstrap samples, each of them obtained by random sampling with replacement N instances from the original dataset.

The accuracy ϵ_{boot} is then estimated as

$$\epsilon_{\mathrm{boot}} = 0.632\epsilon_b + 0.368\epsilon_r$$

where ϵ_r is the re-substitution accuracy, and e_b is the accuracy on the bootstrap subset. Multiple bootstrap subsets $D_{b,i}$ must be generated, the corresponding values being used to estimate the accuracy by averaging the results:

$$\bar{\epsilon}_{\mathrm{boot}} = \frac{1}{n_\epsilon} \sum_{i=1}^{n_\epsilon} \epsilon_{\mathrm{boot}}(D_{b,i})$$

and its variance.

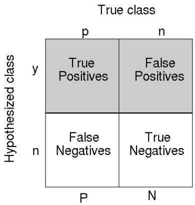
Cross validation

k-fold cross validation is based on the subdivision of the dataset into k mutually exclusive subsets of (approximately) equal size: each one of them is used in turn for testing while the remaining $k-1$ groups are given to the inducer to estimate the parameters of the classifier. If we denote with $D_{\{i\}}$ the set that includes instance i

$$\hat{\epsilon}_k = \frac{1}{N} \sum_i \delta[J(D \setminus D_{\{i\}}; \boldsymbol{x}_i), l_i]$$

Complete cross validation would require averaging over all $\binom{N}{N/k}$ possible choices of the N/k testing instances out of N and is too expensive with the exception of the case $k = 1$ which is also known as leave-one-out (LOO).

ROC representation



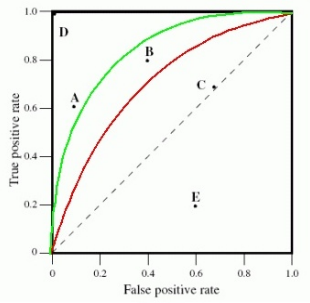
		True class	
		p	n
Hypothesized class	y	True Positives	False Positives
	n	False Negatives	True Negatives
		P	N

The ROC curve describes the performance of a classifier when varying the Neyman-Pearson constraint on P_F:

$$P_D = f(P_F) \quad \text{or} \quad T_{\mathrm{p}} = f(F_{\mathrm{p}})$$

ROC diagrams are not affected by class skewness, and are invariant also to error costs.

ROC points and curves



ROC convex hull

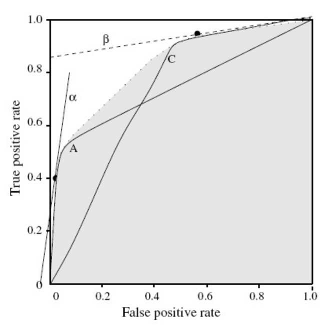
The expected cost of a classifier can be computed from its ROC coordinates:

$$\hat{C} = p(\mathrm{p})(1 - T_{\mathrm{p}})C_{\eta\mathrm{p}} + p(\mathrm{n})F_{\mathrm{p}}C_{\pi\mathrm{n}}$$

Proposition

For any set of cost $(C_{\eta\mathrm{p}}, C_{\pi\mathrm{n}})$ and class distributions $(p(\mathrm{p}), p(\mathrm{n}))$, there is a point on the ROC convex hull (ROCCH) with minimum expected cost.

Operating conditions



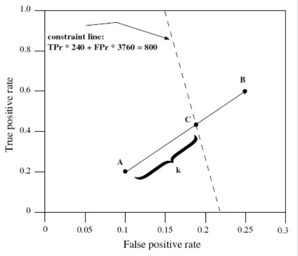
ROC interpolation

Proposition

ROC convex hull hybrid *Given two classifiers J_1 and J_2 represented within ROC space by the points $\boldsymbol{a}_1 = (F_{\mathrm{p}1}, T_{\mathrm{p}1})$ and $\boldsymbol{a}_2 = (F_{\mathrm{p}2}, T_{\mathrm{p}2})$, it is possible to generate a classifier for each point $\boldsymbol{a}_x$ on the segment joining $\boldsymbol{a}_1$ and $\boldsymbol{a}_1$ with a randomized decision rule that samples J_1 with probability*

$$p(J_1) = \frac{\|\boldsymbol{a}_2 - \boldsymbol{a}_x\|}{\|\boldsymbol{a}_2 - \boldsymbol{a}_1\|}$$

Satisfying operating constraints



AUC

The area under the curve (AUC) gives the probability that the classifier will score, a randomly given positive instance higher that a randomly chosen one. This value is equivalent to the Wilcoxon rank test statistic W

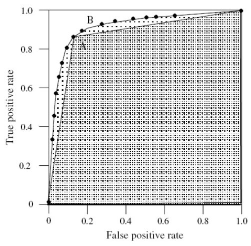
$$W = \frac{1}{N_P N_N} \sum_{i:l_i=\mathrm{p}} \sum_{j:l_j=\mathrm{n}} w(s(\boldsymbol{x}_i), s(\boldsymbol{x}_j))$$

where, assuming no ties,

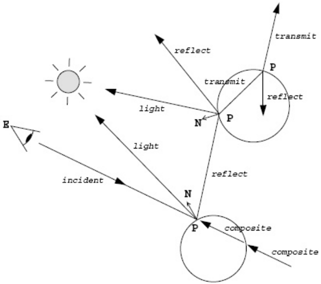
$$w(s(\boldsymbol{x}_i), s(\boldsymbol{x}_j)) = 1 \quad \text{if} \quad s(\boldsymbol{x}_i) > s(\boldsymbol{x}_j)$$

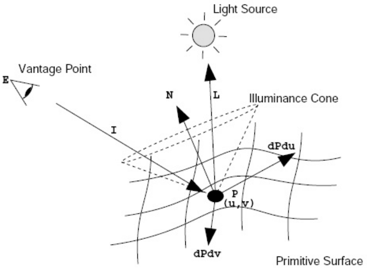
The closer the area to 1, the better the classifier.

Scoring classifiers



Rendering





The appearance of a surface point is determined by solving the rendering equation:

$$L_o(\boldsymbol{x}, -\hat{\boldsymbol{I}}, \lambda) = L_e(\boldsymbol{x}, -\hat{\boldsymbol{I}}, \lambda) + \int_\Omega f_r(\boldsymbol{x}, \hat{\boldsymbol{L}}, -\hat{\boldsymbol{I}}, \lambda) L_i(\boldsymbol{x}, -\hat{\boldsymbol{L}}, \lambda)(-\hat{\boldsymbol{L}} \cdot \hat{\boldsymbol{N}}) d\hat{\boldsymbol{L}}$$

Describing reality: RenderMan®

```
Projection "perspective" "fov" 35
WorldBegin
  LightSource "pointlight" 1 "intensity" 40 "from" [4 2 4]
  Translate    0 0 5
  Color        1 0 0
  Surface      "roughMetal" "roughness" 0.01
  Cylinder     1 0 1.5 360
WorldEnd
```

A simple shader

```
color  roughMetal(normal Nf;  color basecolor;
                          float Ka, Kd, Ks, roughness;)
{
    extern vector I;
    return basecolor * (Ka*ambient() + Kd*diffuse(Nf) +
                        Ks*specular(Nf,-normalize(I),
                                    roughness));
}
```

How realistic is it?

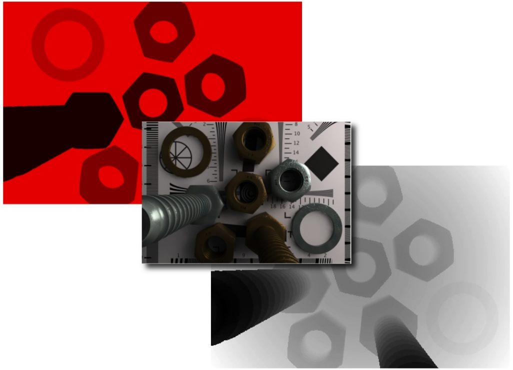
- basic phenomena, including straight propagation, specular reflection, diffuse reflection (Lambertian surfaces), selective reflection, refraction, reflection and polarization (Fresnel's law), exponential absorption of light (Bouguer's law);
- complex phenomena, including non-Lambertian surfaces, anisotropic surfaces, multilayered surfaces, complex volumes, translucent materials, polarization;
- spectral effects, including spiky illumination, dispersion, inteference, diffraction, Rayleigh scattering, fluorescence, and phosphorescence.

Thematic rendering

We can shade a pixel so that its color represents

- the temperature of the surface,
- its distance from the observer,
- its surface coordinates,
- the material,
- an object unique identification code.

Automatic ground truth



References

R. Brunelli and T. Poggio, 1997, Template matching: Matched spatial filters and beyond. *Pattern Recognition* **30**, 751–768.

R. Brunelli, 2009 *Template Matching Techniques in Computer Vision: Theory and Practice*. J. Wiley & Sons

T. Moon and W. Stirling, 2000 *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall.

J Piper, I. Poole and A. Carothers A, 1994, Stein's paradox and improved quadratic discrimination of real and simulated data by covariance weighting *Proc. of the 12th IAPR International Conference on Pattern Recognition (ICPR'94)*, vol. 2, pp. 529–532.