Image Enhancement Algorithms for Night Vision Images

DIGITAL IMAGE PROCESSING

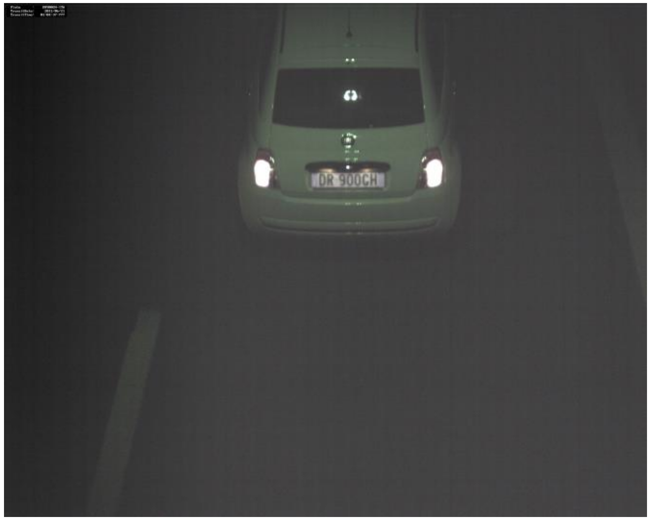
A.A. 2013/14

Facoltà di Ingegneria
Università degli Studi di Brescia

Eridia Bozzetti 91787 Davide Rocco 85422

Introduction

Fields of engagement: traffic



Introduction

In systems of reading motor vehicle license plate is frequent to use a **dual-head system**:

- ✓ The first head performs the reading of license plate
- ✓ The second head is used as the camera of the context in order to have an image with a wider field of view which documents the passage of the vehicle and possibly allows you to make out the vehicle shape

Main problems:

Night vision license plate

View the vehicle shape

Techniques

- Histogram equalization
- Ahe - Adaptive Histogram Equalization
- Clahe - Contrast limited Ahe
- Retinex
- ACE

Histogram equalization

Histogram equalization

- the histogram of an image is a discrete function:

$$h(k) = \mathrm{n}_k \,/(MxN)\,, \quad k \in [0, \dots L-1]$$

where:
L is the number of values that can be assumed by each pixel (from 0 to 255)
n_k is the number of pixels with intensity k
MxN is the number of image pixels

- The function $h(k)$ ($\sum_k h(k) = 1$) is an estimation of the probability distribution of the pixel intensity

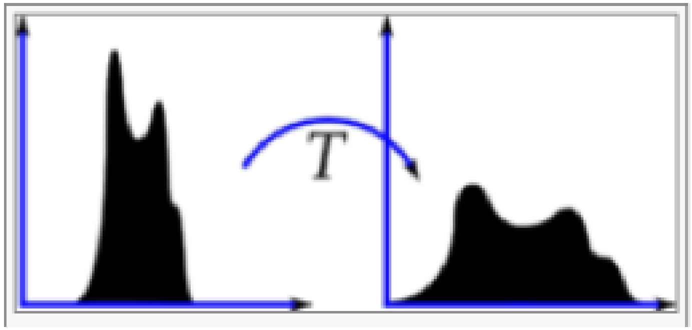
$$F(k) = \sum_{i=0}^{k} h(i)$$ is the cumulative function

- The Histogram provides information about statistical image properties useful for contrast enhancement

Histogram equalization - algorithm

algorithm to equalize a grayscale image using histogram:

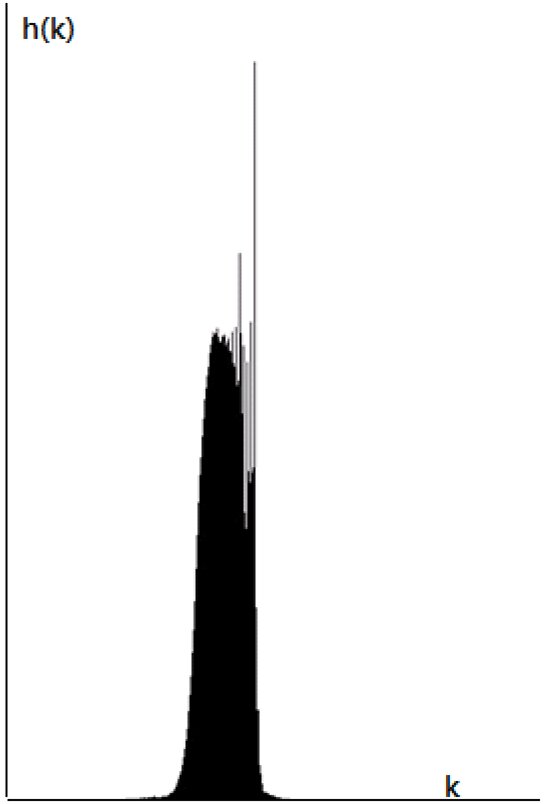
1- compute the image histogram $h(k)$
2- compute the cumulative function $F(k)$
3- apply trasformation $I_{eq}(i,j) = T[I(i,j)] = F[I(i,j)]$
4- Rescale $I_{eq}[i,j]$ form [0 - 1] to [0,...,L - 1]



Advantages: it is very simple
Disadvantages: it doesn't take account of local image information

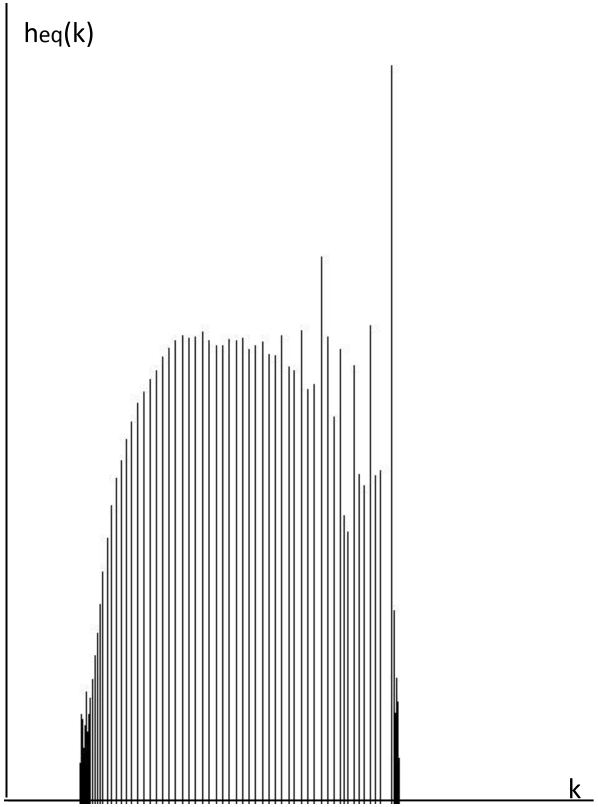
Histogram equalization - examples

✓ Useful for homogeneous image
Low contrast image have a narrow histogram:



Histogram equalization - examples

- Image after equalization



Histogram equalization - Results

Histogram equalization - Results

This method is useful only if we use homogeneous images. When we apply this method to our images the results are not satisfied:

Adaptive Histogram Equalization

AHE - adaptive histogram equalization

- ✓ It differs from ordinary histogram equalization
- ✓ computes several histograms each corresponding to a distinct section of the image
- ✓ suitable for improving the local contrast of an image and bringing out more detail

The traditional AHE algorithm can be expressed as in Algorithm 1
we assume the square contextual region with block size W^2 in AHE

Algorithm 1 Traditional AHE

for every pixel *i* (with grey level *l*)in image **do**
Initialize array *Hist* to zero;
    for every contextual pixel *j* **do**
$$Hist[g(j)] = Hist[g(j)] + 1$$
    end
Sum: $CHist_l = \sum_{k=0}^{l} Hist(k)$
$$l' = CHist_l * L/W^2$$
end

AHE - adaptive histogram equalization

we could find that AHE is quite computationally expensive:

- ✓ For every pixel, it need W^2 additions to get the local histogram, and I additions for *CHistI*, one multiplication and one division to map the origin grey level to new one

- ✓ For an image with size $M*N$, AHE's computation complexity will be $O(M*N*W^2)$, when the image size and block size become large, the computation time becomes unbearable

- ✓ the expensive computation complexity prohibits it to be used in real-time occasion

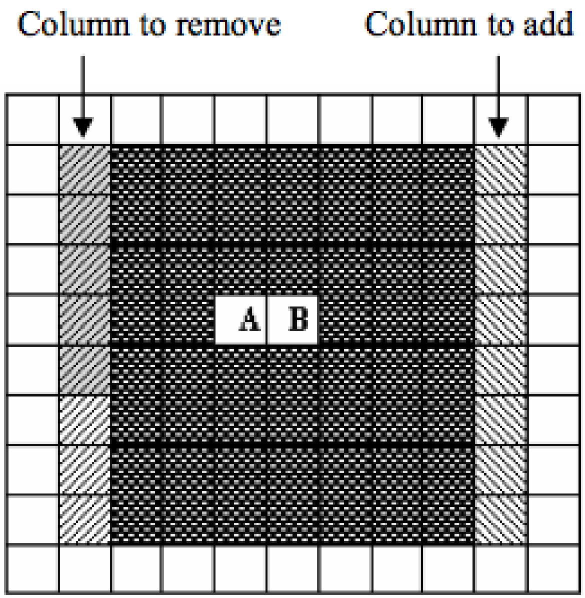
FAST AHE

FAST - AHE

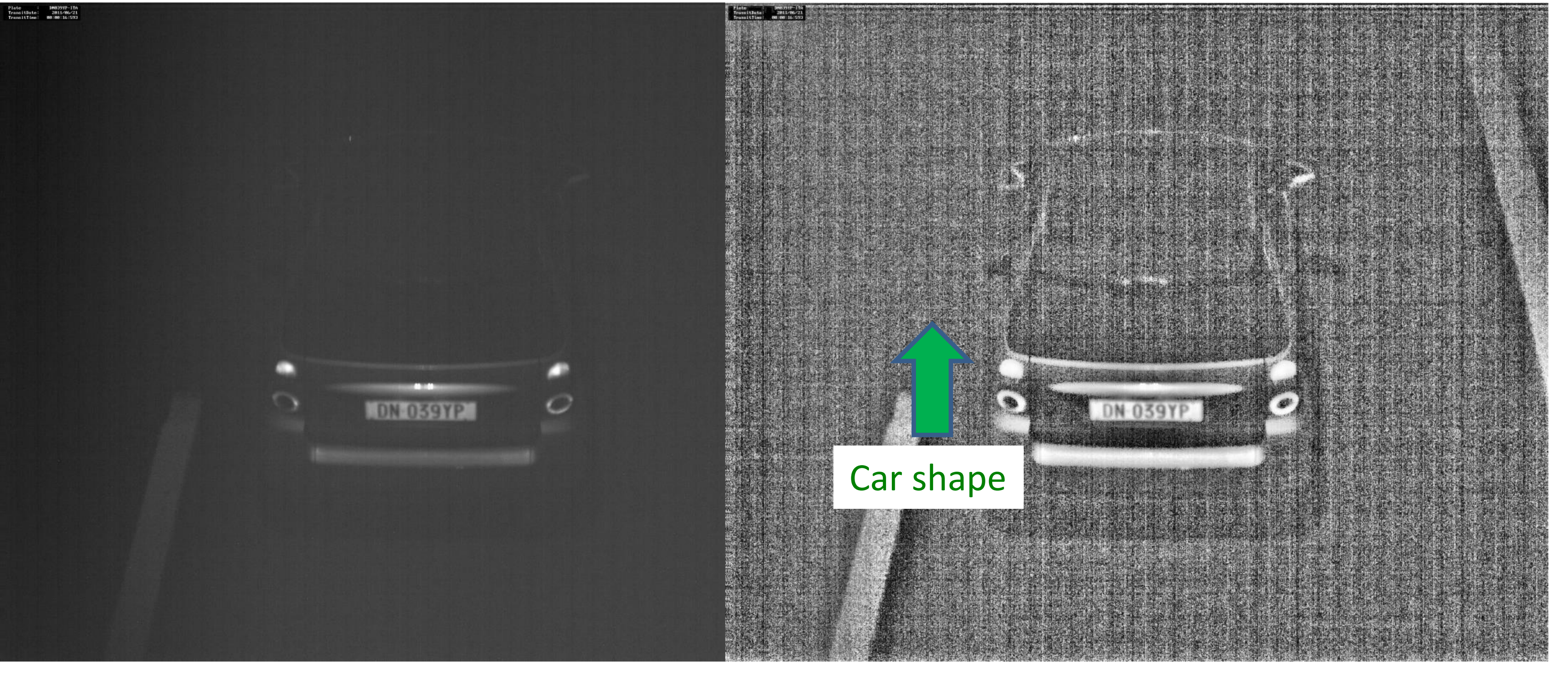
to reduce the computation:

- ✓ when window center moves from A to B, in order to obtain the histogram of the next block, we need not re-scan the entire contextual region
- ✓ we can just remove the left column pixels of last block from current histogram and add the right column of current block to it

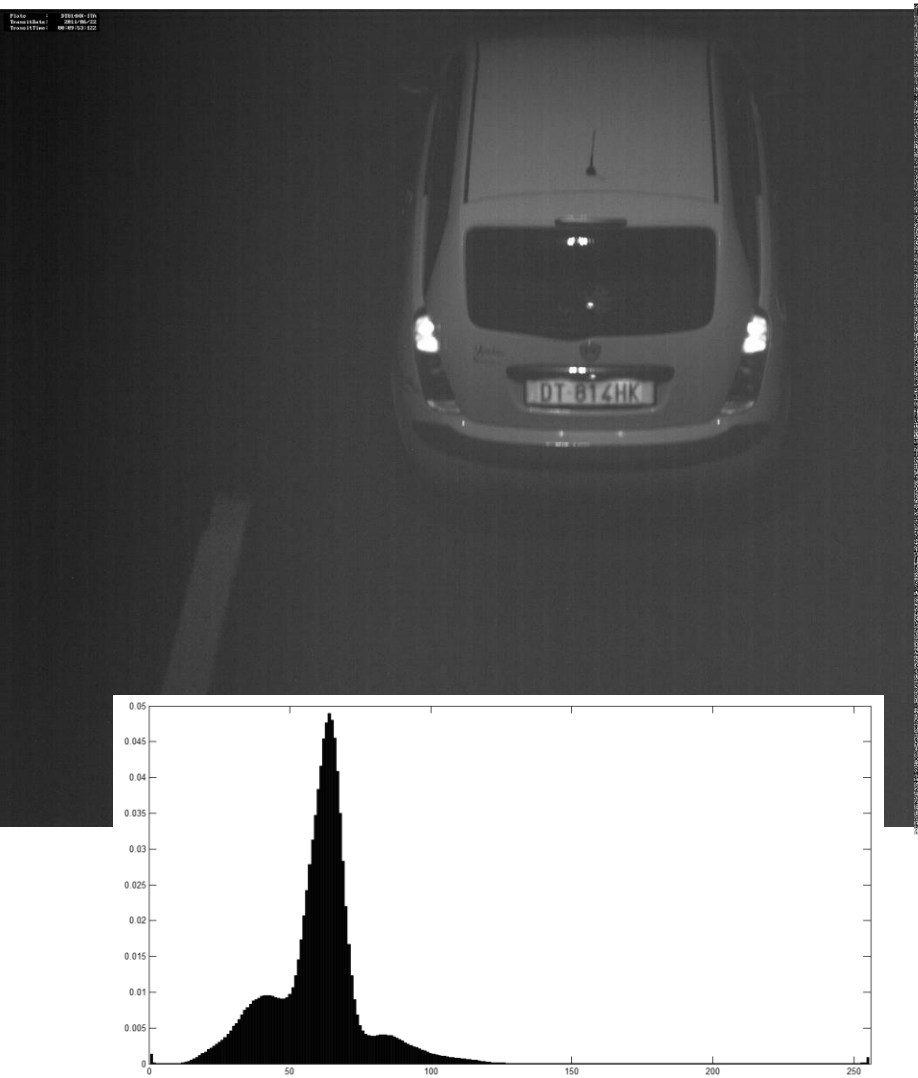
Only the first block in an image needs to process every pixel in the block

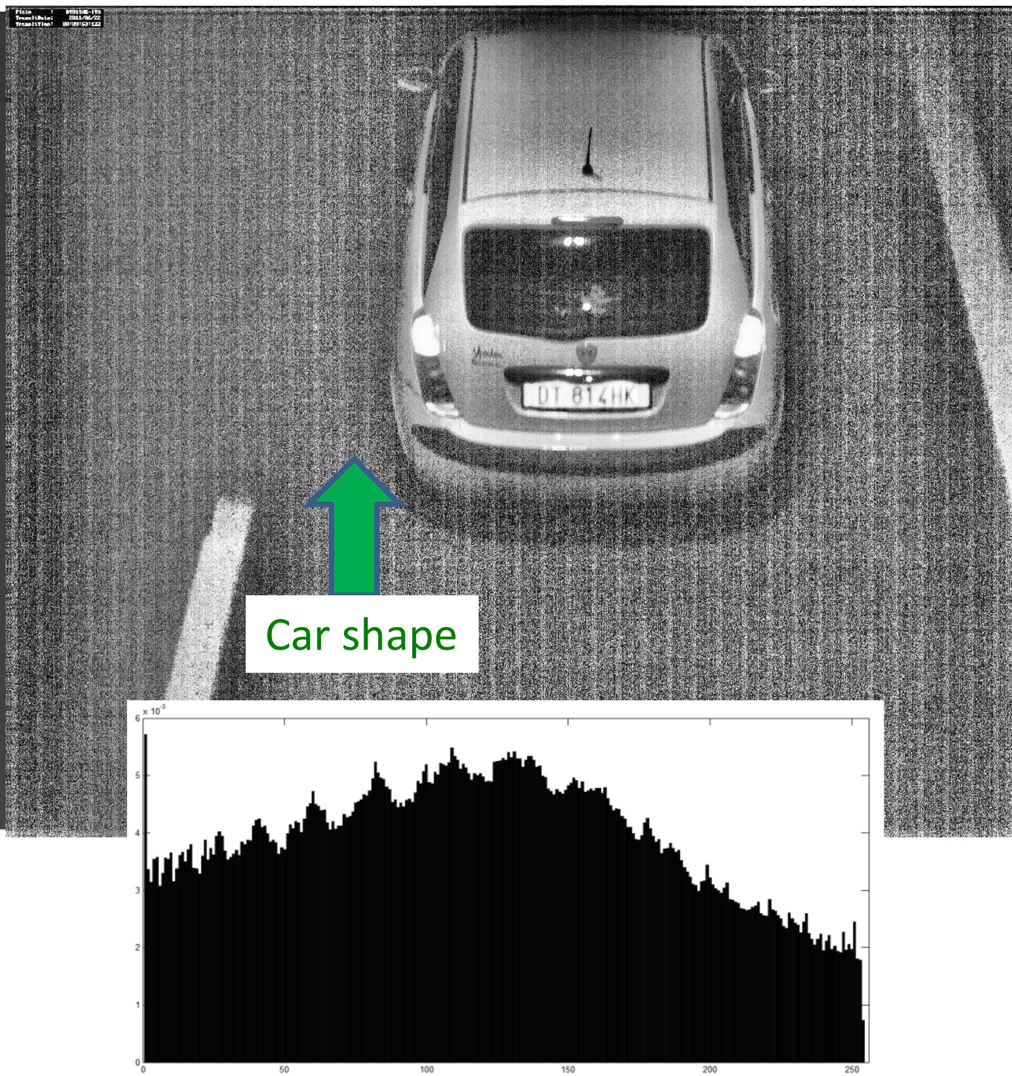


AHE - adaptive histogram equalization



AHE- adaptive histogram equalization





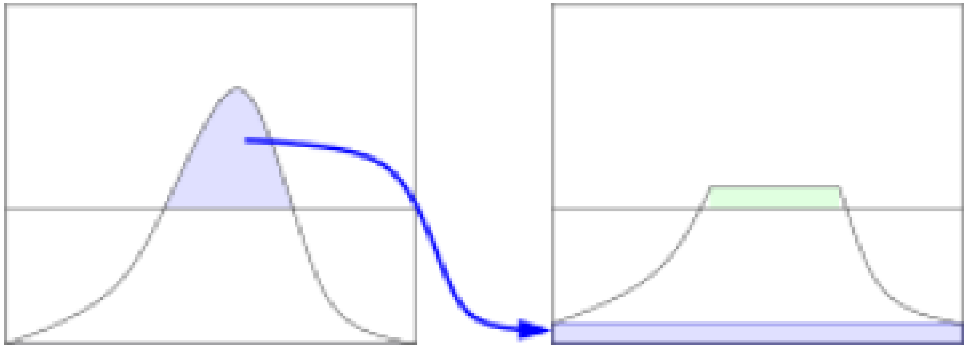
Clahe - Contrast limited Ahe

Clahe - Contrast Limited Adaptive Histogram Equalization

- Contrast Limited AHE differs from ordinary adaptive histogram equalization in its contrast limiting
- it was developed to prevent the overamplification of noise that adaptive histogram equalization can give rise to

- by clipping the histogram at a predefined value before computing the cumulative density function. This limits the slope of the cumulative density function and therefore of the transformation function

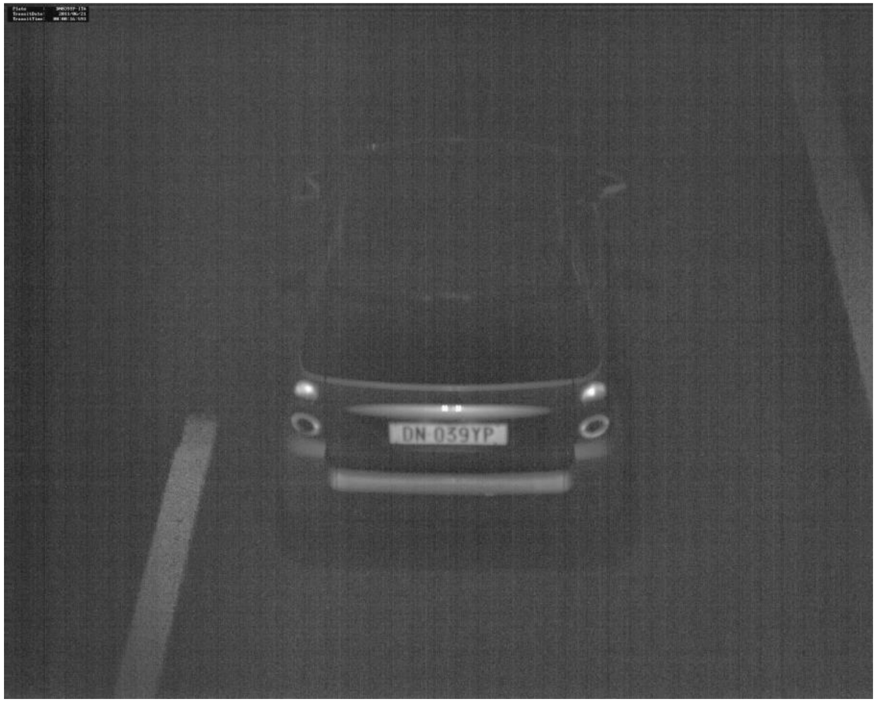
Clahe - Contrast Limited Adaptive Histogram Equalization



- The value at which the histogram is clipped, the so-called clip limit, depends on the normalization of the histogram and thereby on the size of the neighbourhood region
- It is advantageous not to discard the part of the histogram that exceeds the clip limit but to redistribute it equally among all histogram bins

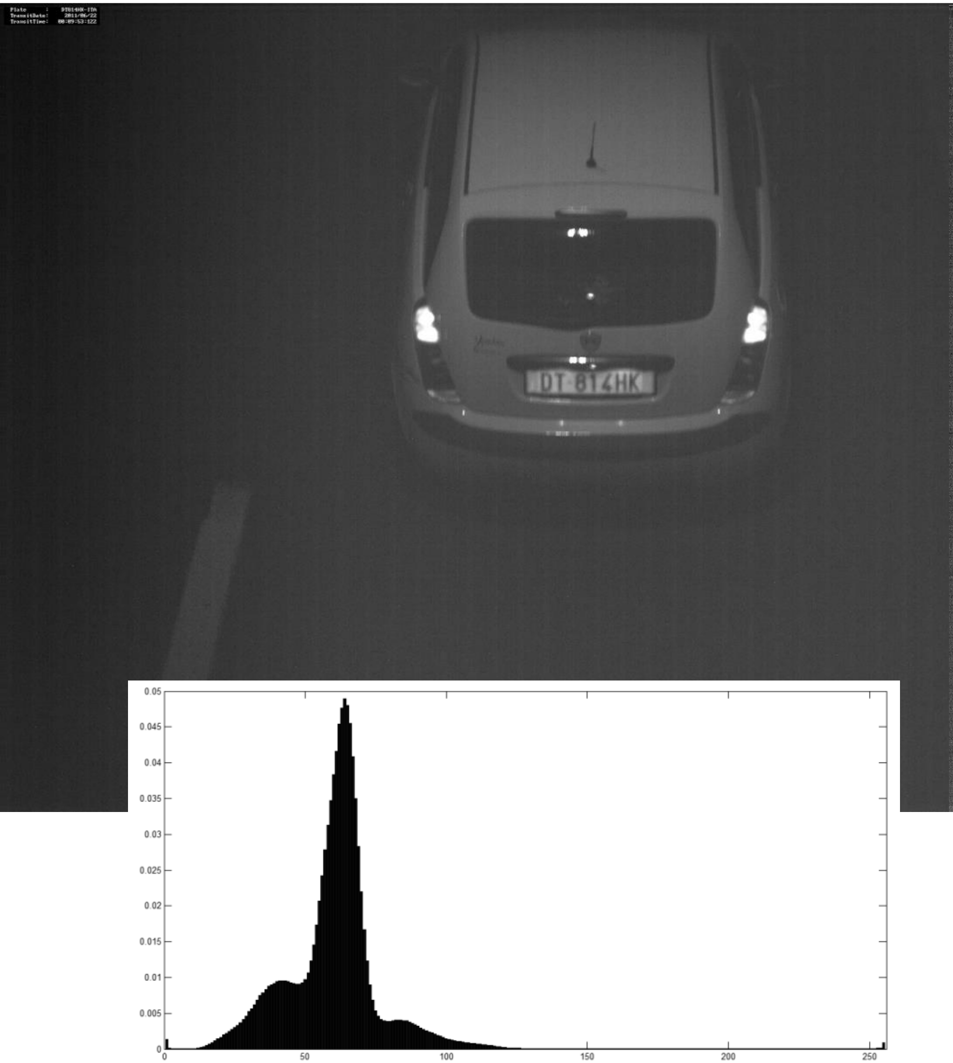
Clahe - Contrast Limited Adaptive Histogram Equalization

Clipping level = 0.1



Clipping level = 0.02

Clahe - Contrast Limited Adaptive Histogram Equalization



RETINEX

Assumption

- ✓ **Retinex is a simplified model of the HVS**

- ✓ The perceived color of a unit area is determined by the relationship between this unit area and the rest of unit areas in the image, indipendently in each wave-band, and does not depend on the absolute value of light

- ✓ There is a quantity called **"lightness"** which is **associated to the objects regardless of changes in the illumination or in the position of the objects in the scene**

Retinex Algorithm

- ✓ The **lightness information** is **estimated** by computing **sequential ratios between values at adjacent points** of a series of random paths in the image

- ✓ **Changes above a certain threshold are considered as changes in reflectance.** If instead color changes are smaller than the threshold they are considered as illumination changes, and the current ratio is set to one

- ✓ After computing on many paths, the result on each path is averaged to obtain the lightness

Retinex Algorithm

The image data *I(x)* is the intensity value for each chromatic channel at *x*.

Consider a collection of *N* paths $\Upsilon_1, \ldots, \Upsilon_N$ starting at j_k and ending at x.

Let n_k be the number of pixels of the path Υ_k and denote by $x_{tk} = \Upsilon_k(t_k)$ for $t_k = 1, \ldots, n_k$ and by $x(t_k+1) = \Upsilon_k(t_k + 1)$ the subsequent pixel of the path.

The lightness value $L(x)$ of a pixel *x* in a given chromatic channel is the average of the relative lightness at x over all paths, that is

$$L(x) = \frac{\sum_{k=1}^{N} L(x; j_k)}{N} \ ,$$

where $L(x; j_k)$ denotes the relative lightness of a pixel *x* with respect to j_k defined by

$$L(x; j_k) = \sum_{t_k=1}^{n_k} \delta \left[\log \frac{I(x_{t_k+1})}{I(x_{t_k})} \right] \ ,$$

and, for a fixed **threshold t**,

$$\delta(s) = \begin{cases} s & \text{if } |s| > t \\ 0 & \text{if } |s| < t \ . \end{cases}$$

Fast Implementation of Retinex

Retinex formalized as a Poisson equation.

Define:

$$F(x) = f\left(\frac{I(x)}{I(x_{-0})}\right) + f\left(\frac{I(x)}{I(x_{+0})}\right) + f\left(\frac{I(x)}{I(x_{0-})}\right) + f\left(\frac{I(x)}{I(x_{0+})}\right) .$$

where x_{-0}, x_{+0}, x_{0-} and x_{0+} represent the four discrete x-neighbors, and f(x) = δ (log x)

The lightness value in a chromatic channel L is the unique solution of the discrete Poisson equation with Neumann boundary condition,

$$\begin{cases} -\Delta_d L(x) = F(x) & x \in \Omega \\ \frac{\partial L}{\partial n} = 0 & x \in \partial\Omega \end{cases} ,$$

If we take δ (s) = s then the function F becomes Δ_d log(I(x)) and the equation becomes

$$\Delta_d L(x) = \Delta_d \log(I(x))$$

The Poisson equation can be solved using the Fourier transform.

$$\mathcal{F}[\Delta_d L(x)] = \mathcal{F}[\Delta_d \log(I(x))]$$

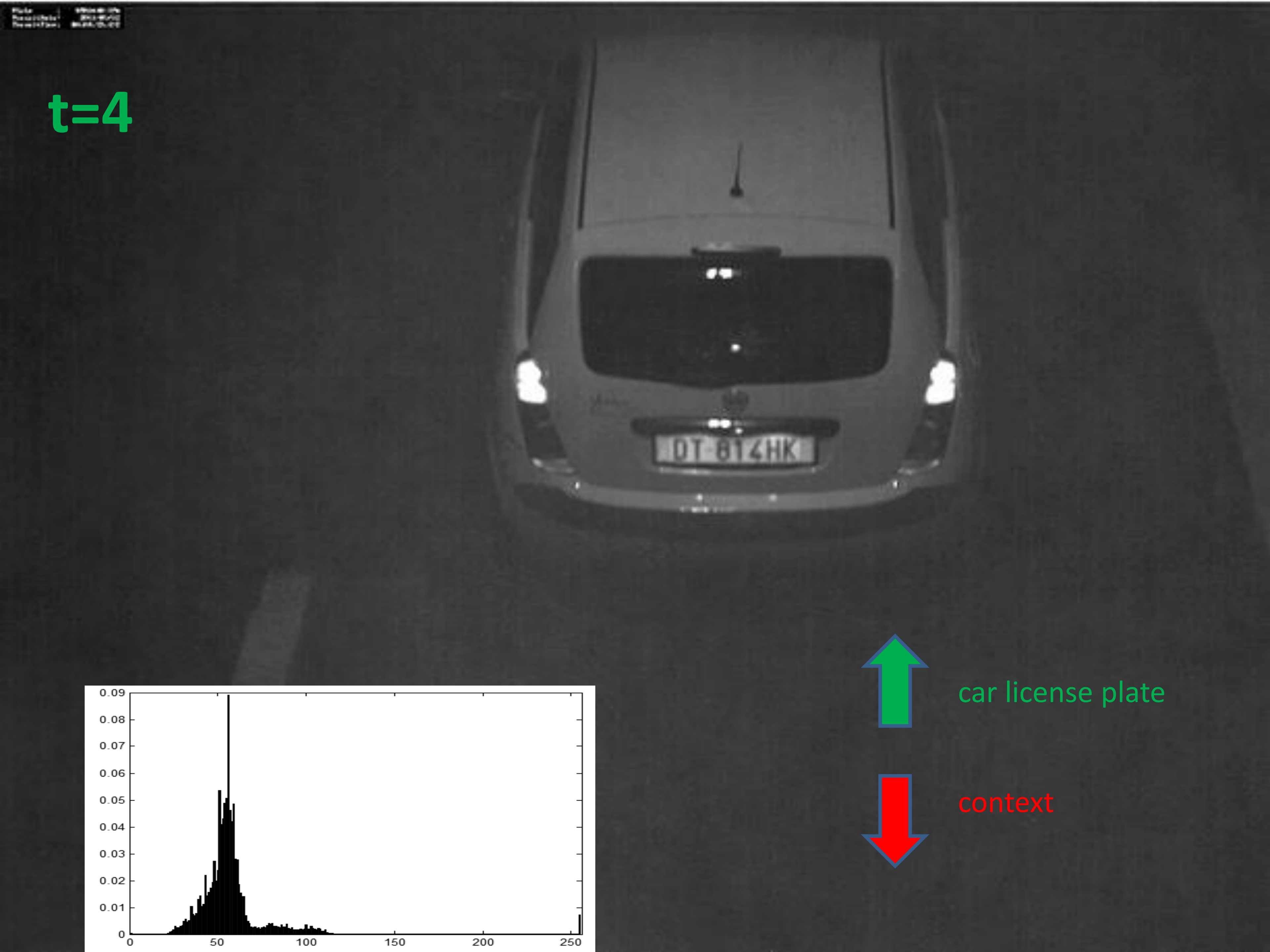
Fast Implementation of Retinex

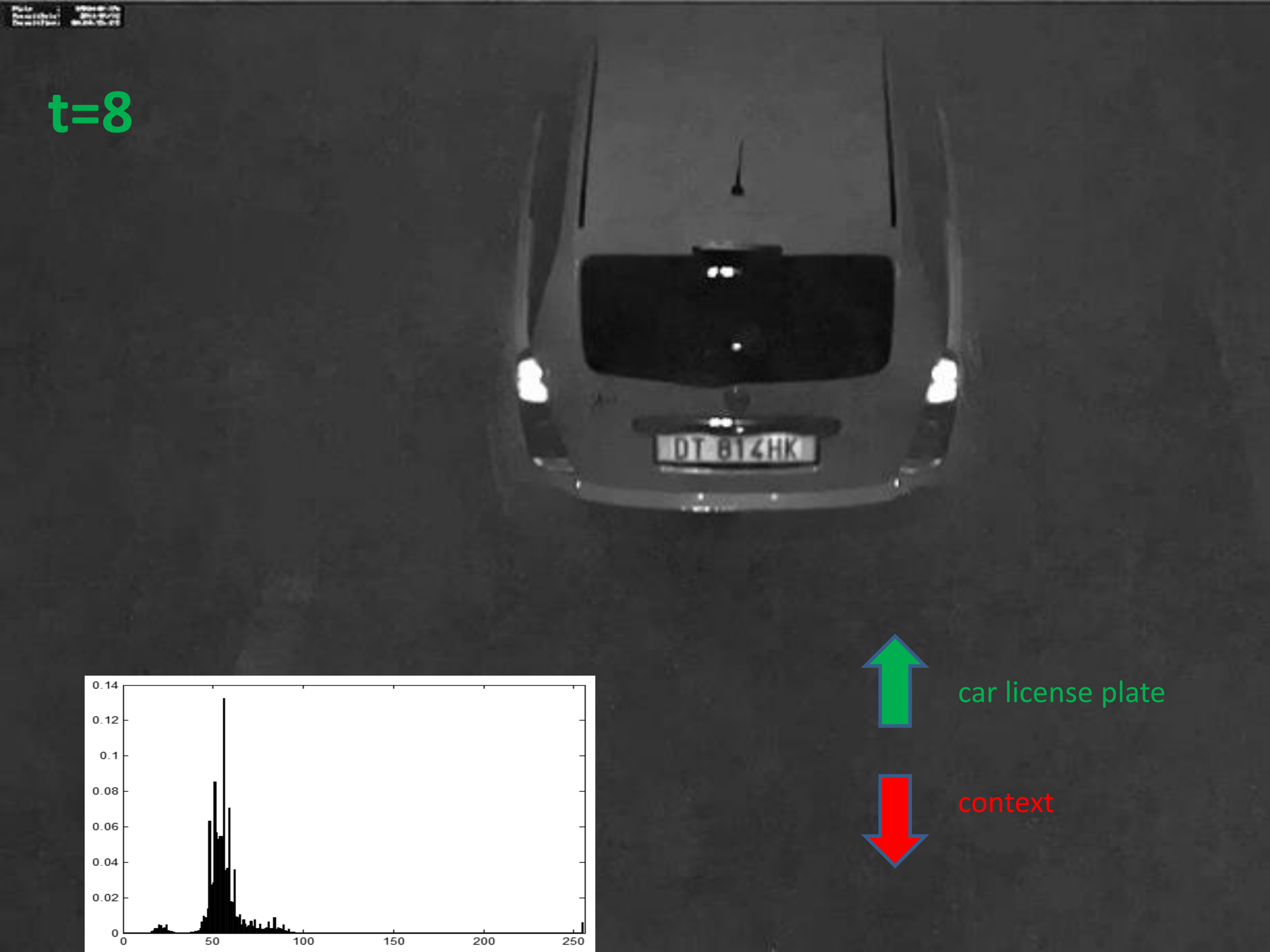
$$\widehat{L}(k,l) = \begin{cases} \dfrac{\widehat{F}(k,l)}{4-2\cos\frac{2\pi k}{N}-2\cos\frac{2\pi l}{M}} & \text{if } (k,l) \neq (0,0) \\ 0 & \text{if } (k,l) = (0,0) \end{cases}.$$

The algorithm (applied to each channel) therefore is:

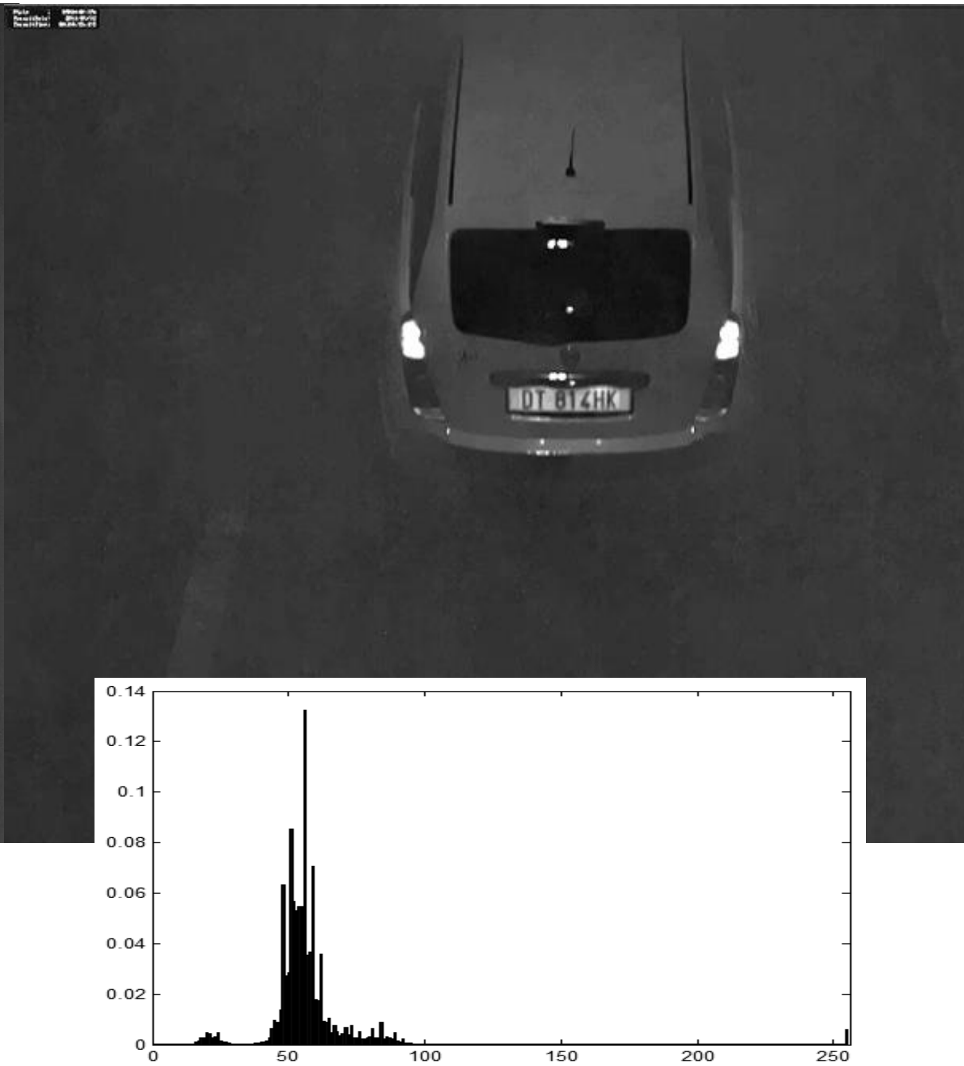
1. Compute *F(i,j)*;
2. Compute the Fourier transform of *F* by DFT
3. Deduce the Fourier transform of *L* using the formula above;
4. Compute the final solution *L* by the inverse DFT and apply the normalization.


DT 814HK
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250


t=4
DT 814HK
car license plate
context
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
50
100
150
200
250


t=8
DT 814HK
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
100
150
200
250
car license plate
context

Retinex - Results



ACE

Automatic Color Enhancement

Introduction

- The ACE method is based on a simple model of the human visual system.

- It's inspired by several low level mechanisms:

  -gray world: the average perceived color is gray

  -white patch: normalization toward a white reference

  -lateral inhibition

- The enhanced image appears natural because the input image is adjusted in a manner consistent with perception.

THE IDEA

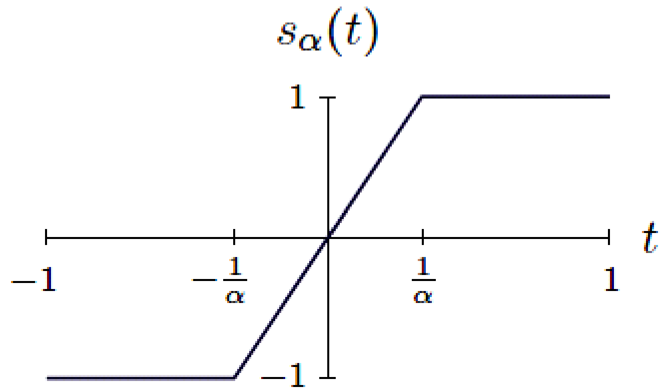
I = input grayscale image with domain Ω and intensity values scaled in [0, 1].

For a color image, the following operation is performed,independently on the RGB channels:

$$R(x) = \sum_{y \in \Omega \setminus x} \frac{s_\alpha(I(x) - I(y))}{\|x - y\|}, \quad x \in \Omega,$$

1° STAGE

Where : Ω\x denotes {y ∈ Ω : y ≠ x},
‖x-y‖ denotes Euclidean distance,
s_α : [−1, 1] → R is the slope function $s_\alpha(t)$ = min{max{αt,−1}, 1 } for some α ≥ 1.



In the limit α→ ∞, it is the signum function s_α(t)=sign(t).

THE IDEA

The enhanced channnel is computed by stretching R to [0,1] as

$$L(x) = \frac{R(x) - \min R}{\max R - \min R}.$$

2° STAGE

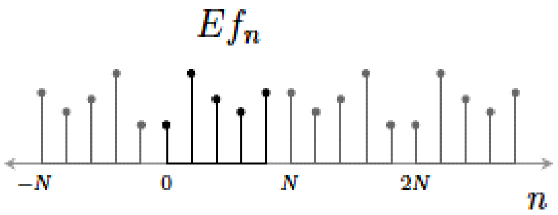
The first stage of the method adapts local image contrast.
Lateral inhibition is simulated by neighbor differences $I(x) - I(y)$ and weighting according to distance $\|x - y\|$.
The function s_α amplifies small differences and saturates large differences, which has the effect of expanding or compressing the dynamic range according to the local image content.

The second stage adapts the image to obtain a global white balance.
By implementing these mechanisms, ACE is a simplified model of the human visual system: the enhancement process is consistent with perception.

Boundary Handling and Convolutions

Define the half-sample symmetric extension *Ef* of an N-sample sequence *f*

$$Ef_n = \begin{cases} f_n & \text{if } n = 0, \ldots, N-1, \\ Ef_{-1-n} & \text{if } n < 0, \\ Ef_{2N-1-n} & \text{if } n \geq N. \end{cases}$$



Ef is 2N-periodic,

The domain can be interpreted to be a circle of 2N samples.

1 D

The tensor product of this extension applied to an N × N image $u_{i,j}$, $i = 0, \ldots, N-1$, $j = 0, \ldots, N-1$

In 2D, the domain is the $2N \times 2N$-periodic torus T_2.

For any $x, y \in T_2$, distance is defined on the torus as

$$d(x,y) := \min_{\bar{x},\bar{y}} \{|\bar{x} - \bar{y}| : \bar{x} \equiv x, \bar{y} \equiv y\},$$

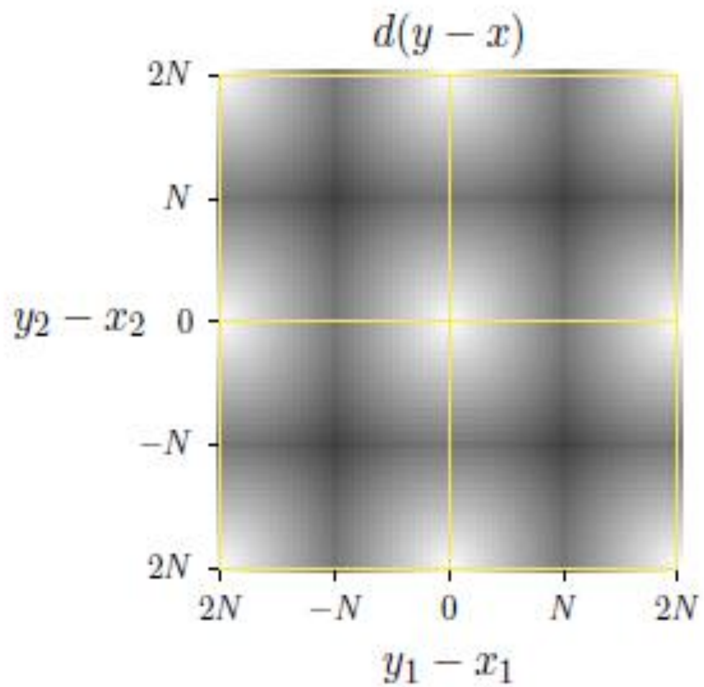
where $|v| := \sqrt{v_1^2 + v_2^2}$ and $\equiv$ denotes equivalence on torus.

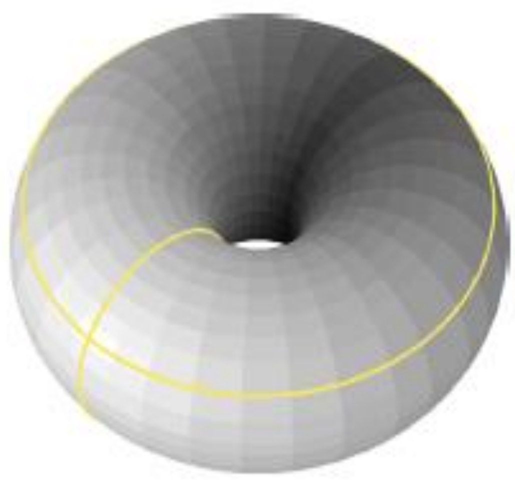
2 D

Boundary Handling and Convolutions

The summation R(x) is redefined as a summation over the torus *T2\x*, and Euclidean distance ‖*x* − *y*‖ is replaced by torus distance *d(x − y)*:

$$R(x) = \sum_{y \in \mathbb{T}^2 \setminus x} \frac{s_\alpha\big(I(x) - I(y)\big)}{d(x-y)}, \quad x \in \Omega.$$





Boundary Handling and Convolutions

Defining:

$$\omega(x-y) = \begin{cases} 0 & \text{if } x = y, \\ 1/d(x-y) & \text{if } x \neq y, \end{cases}$$

we compute R as:

$$R(x) = \sum_{y \in \mathbb{T}^2} \omega(x-y) s_\alpha\big(I(x) - I(y)\big).$$

Both algorithms (**Polynomial Slope Function** and **Interpolation**) will approximate R in terms of convolutions with ω on T^2.

1.Polynomial Approximation

The key change to the ACE method is to approximate *min{max{αt,−1}, 1}* with an odd polynomial approximation:

$$s_\alpha(t) \approx \sum_{m=1}^{M} c_m t^m$$

It is then possible to decompose R into a sum of convolutions:

$$R(x) = \sum_{y \in \mathbb{T}^2} \omega(x-y) \sum_{m=1}^{M} c_m \big(I(x) - I(y)\big)^m$$

$$= -\sum_{y \in \mathbb{T}^2} \omega(x-y) \sum_{m=1}^{M} c_m \big(I(y) - I(x)\big)^m$$

$$= -\sum_{y \in \mathbb{T}^2} \omega(x-y) \sum_{m=1}^{M} c_m \sum_{n=0}^{m} \binom{m}{n} I(y)^n \big(-I(x)\big)^{m-n}$$

$$= \sum_{n=0}^{M} \underbrace{\left(\sum_{m=n}^{M} c_n \binom{m}{n} (-1)^{m-n+1} I(x)^{m-n} \right)}_{a_n(x)} \sum_{y \in \mathbb{T}^2} \omega(y-x) I(y)^n$$

$$= \sum_{n=0}^{M} a_n(x) (\omega * I^n)(x),$$ where $*$ is cyclic convolution over T^2.

1.Polynomial Approximation

$$s_{\alpha}(t) \approx \sum_{m=1}^{M} c_m t^m$$

By the Stone–Weierstrass theorem, the continuous function $s_{\alpha}(t)$ can be uniformly approximated on [−1, 1] by a polynomial with any desired precision.

3M convolutions must be evaluated for an M_{th} degree polynomial → compromise is necessary between accuracy and speed.

For a fixed polynomial degree M, we select the coefficients c_m to minimize the maximum absolute error over [−1, 1],

$$\min_{c} \max_{t\in[-1,1]} \left| s_{\alpha}(t) - \sum_{m=1}^{M} c_m t^m \right| .$$

1.Polynomial Approximation

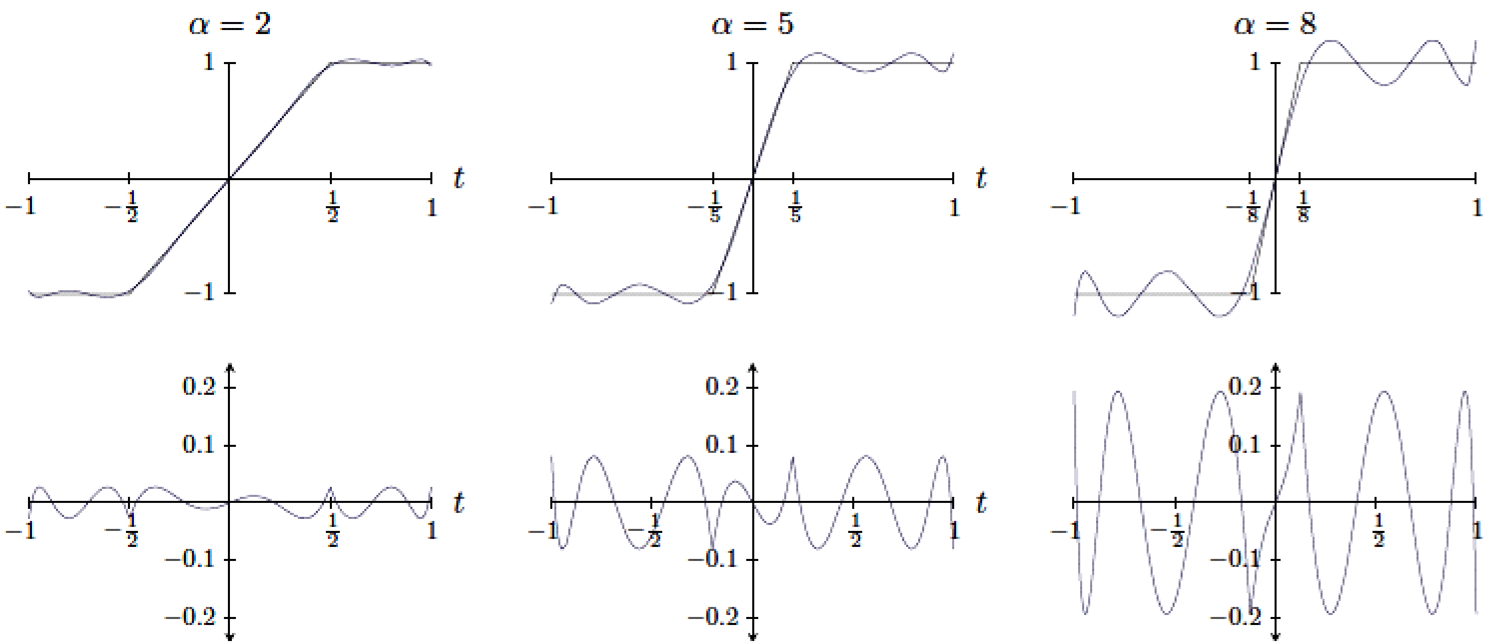
The optimal *c* can be found using the Remez algorithm.

Optimal 9th degree approximation of s_α for different α.

α	Polynomial	Max error
1	t	0.000
2	$1.85623249t + 3.82397125t^3 - 19.70879455t^5 + 26.15510902t^7 - 11.15375327t^9$	0.028
3	$3.51036396t - 6.31644952t^3 + 0.92439798t^5 + 9.32834829t^7 - 6.50264005t^9$	0.057
4	$4.76270090t - 18.23743983t^3 + 36.10529118t^5 - 31.35677926t^7 + 9.66532431t^9$	0.061
5	$5.64305564t - 28.94026159t^3 + 74.52401661t^5 - 83.54012582t^7 + 33.39343065t^9$	0.081
6	$6.19837979t - 35.18789052t^3 + 95.28157108t^5 - 109.95601312t^7 + 44.78177264t^9$	0.118
7	$6.69888108t - 41.02503190t^3 + 115.02784036t^5 - 135.35603880t^7 + 55.81014424t^9$	0.156
8	$7.15179080t - 46.43557440t^3 + 133.54648929t^5 - 159.34156394t^7 + 66.27157886t^9$	0.193

1.Polynomial Approximation

Top row: s_α and its 9th degree approximation. Bottom row: approximation error.



For fixed polynomial degree, the approximation error increases with α.

2.Interpolation

Define the sum:

$$R(x;L) = \sum_{y\in\mathbb{T}^2} \omega(x-y)s_\alpha\big(L - I(y)\big),$$

where I(x) has been replaced by a constant L. Since the argument of s_α now depends only on y, the sum is a convolution. This allows for a fast algorithm to approximate ACE.

Let (Lj) be a sequence such that *min I* = $L_1 < L_2 < \cdots < L_J$ = *max I*, and compute *R(x; Lj), j = 1, . . . , J*.
Then approximate *R(x) = R(x; I(x))* by piecewise linear interpolation

$$R(x) \approx R(x;L_j) + \frac{R(x;L_{j+1}) - R(x;L_j)}{L_{j+1} - L_j}(I(x) - L_j), \quad j \text{ such that } L_j \le I(x) \le L_{j+1}.$$

(L_j) is uniformly spaced,

$$L_j = \min I + (\max I - \min I)\tfrac{j-1}{J-1}$$

Using J = 8 levels provides an accurate approximation for typical images.

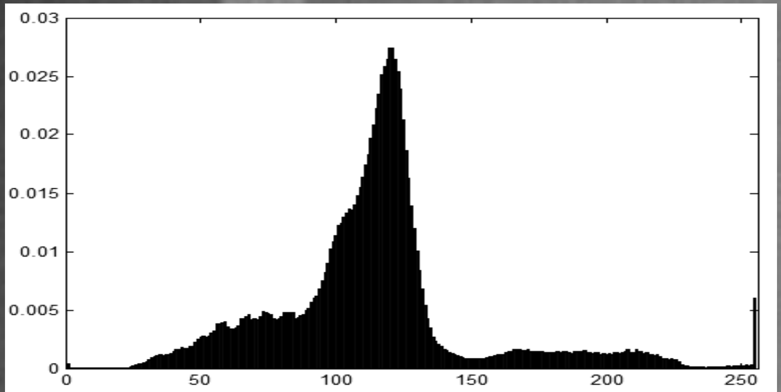
ACE – Results

Plate : DT814HK-ITA
TransitDate: 2011/06/22
TransitTime: 00:09:53:122
DT 814HK
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250

α= 2, omega = 1/r Degree 9 polynomial approximation

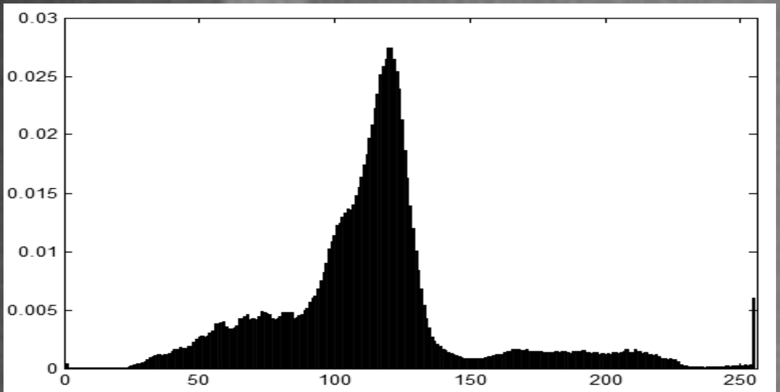
α= 5, omega = 1/r Degree 9 polynomial approximation

Equivalent to interpolation method



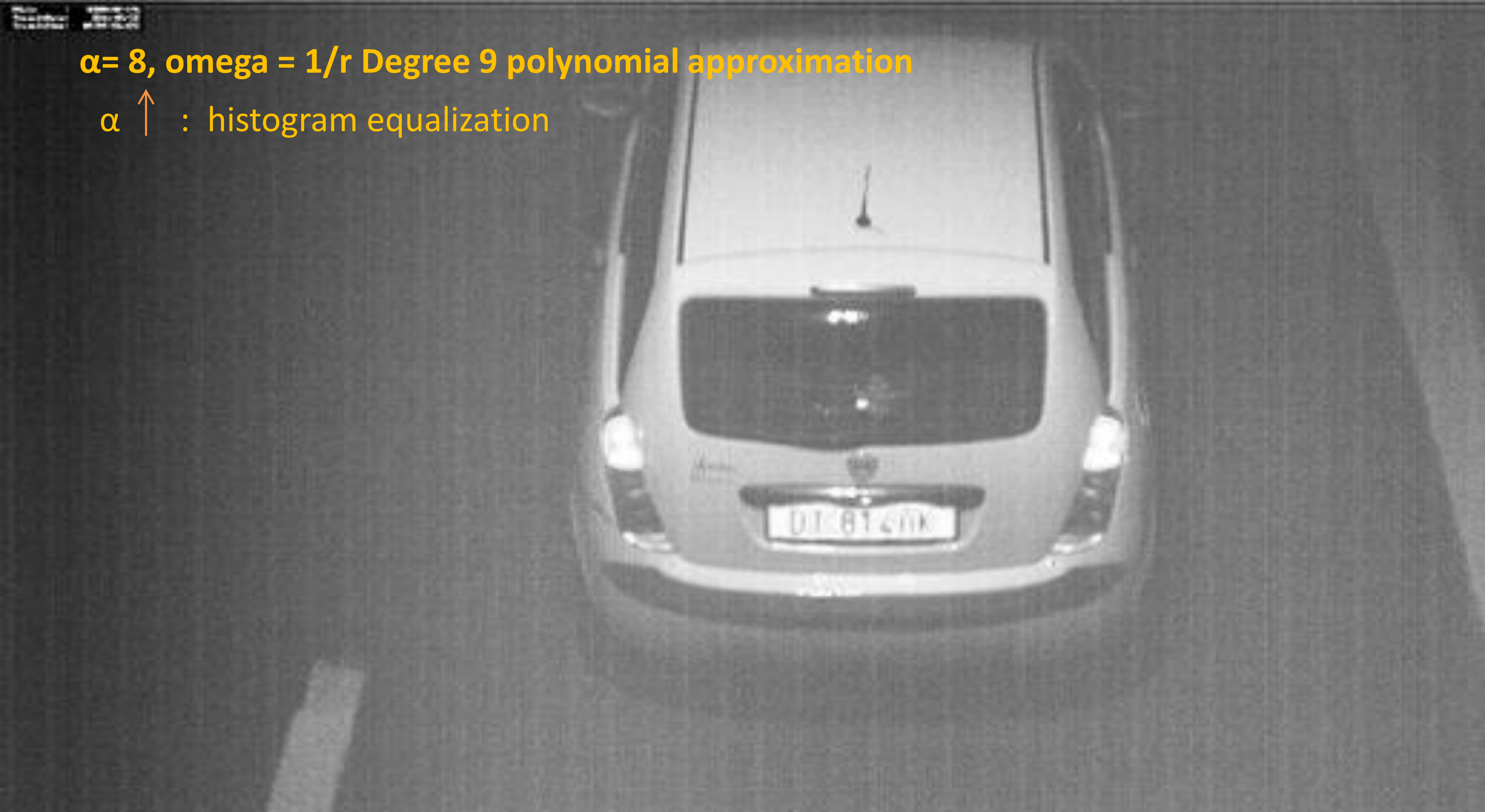
α= 5, omega = 1/r Interpolation with 8 levels

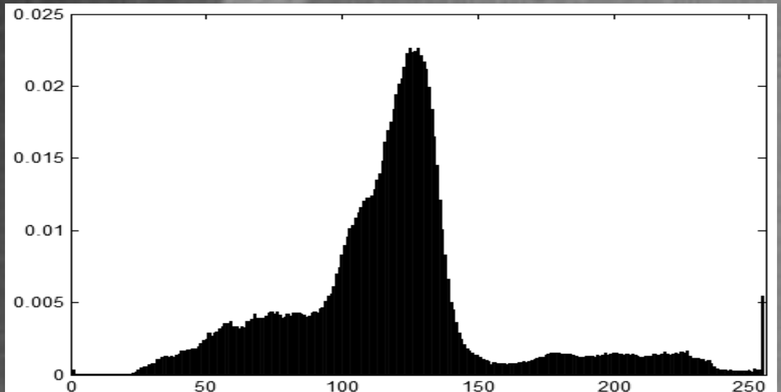
Equivalent to polynomial approximation method



α= 8, omega = 1/r Degree 9 polynomial approximation

α ↑ : histogram equalization





Ace - Results


DT 814HK
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0 50 100 150 200 250

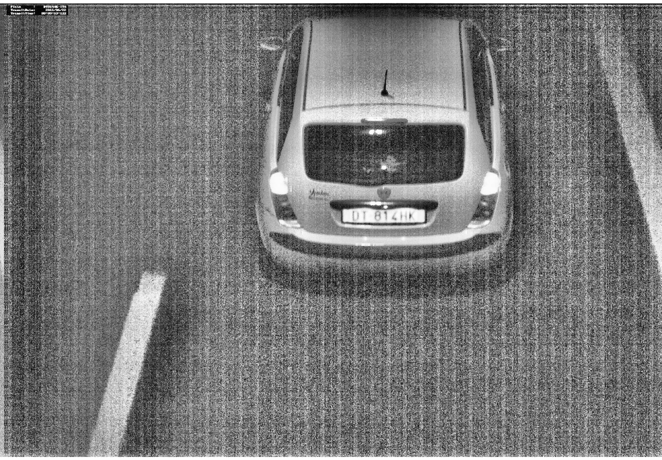

0.025
0.02
0.015
0.01
0.005
0
0 50 100 150 200 250

Comparison

original image

he

ahe



clahe

retinex

ace

Comparison

original image

he

ahe

clahe

retinex

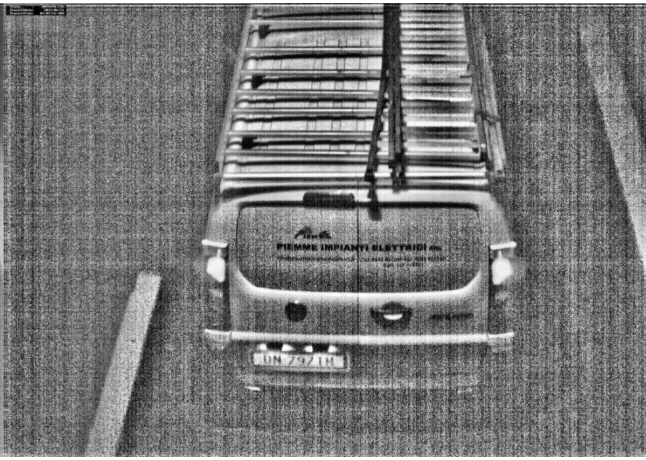
ace

Comparison

original image

he

ahe



clahe

retinex

ace

Thanks for your attention