A Vision-based Fully Automated Approach to Robust Image Cropping Detection

Marco Fanfani[a], Massimo Iuliani[a,b], Fabio Bellavia[a], Carlo Colombo[a,*], Alessandro Piva[a,b]

[a]Dept. of Information Engineering, University of Florence, Florence, Italy
[b]FORLAB Multimedia Forensics Laboratory, University of Florence, Prato, Italy

Abstract

The definition of valid and robust methodologies for assessing the authenticity of digital information is nowadays critical to contrast social manipulation through the media. A key research topic in multimedia forensics is the development of methods for detecting tampered content in large image collections without any human intervention. This paper introduces AMARCORD (**A**utomatic **M**anhattan-scene **A**symmet**R**ically **C**r**O**pped image**R**y **D**etector), a fully automated detector for exposing evidences of asymmetrical image cropping on Manhattan-World scenes. The proposed solution estimates and exploits the camera principal point, i.e., a physical feature extracted directly from the image content that is quite insensitive to image processing operations, such as compression and resizing, typical of social media platforms. Robust computer vision techniques are employed throughout, so as to cope with large sources of noise in the data and improve detection performance. The method leverages a novel metric based on robust statistics, and is also capable to decide autonomously whether the image at hand is tractable or not. The results of an extensive experimental evaluation covering several cropping scenarios demonstrate the effectiveness and robustness of our approach.

Keywords: Multimedia Forensics, Robust Computer Vision, Cropping Detection, Image Content Analysis

[*]Corresponding author
Email address: carlo.colombo@unifi.it (Carlo Colombo)

1. Introduction

Automatic methods able to detect forgeries in digital images are fundamental to counter the ever-increasing production and spread of fake imagery through the media. Image forensic methods [1, 2, 3, 4, 5, 6, 7] try to solve this problem by observing distinctive traces left by manipulation operations. Depending on the exploited evidences, forensic methods can be broadly classified into *signal-based* and *scene-based*. The former look for invisible footprints introduced in the signal statistics, like demosaicing artefacts [8], sensor noise [9], or compression anomalies [10, 11, 12]. Scene-based methods try instead to detect inconsistencies left directly within the elements of the depicted scene, such as shadows [13], lighting [14, 15, 16], or object perspective and geometry [17, 18, 19, 20]. Across the years, a great attention has been devoted to signal-based approaches with interesting results, even in automatic frameworks. Nevertheless, these methods are often ineffective when the investigated content undergoes a processing chain (e.g., filtering, resizing and compression) that may partially or completely spoil the traces left by previous operations [21]. On the other hand, scene-based solutions can cope effortlessly with non-native contents, but they are not popular yet in the forensic domain, as they usually require specific features that are both difficult to detect and prone to noise, thus making it quite arduous to avoid altogether manual intervention. This implies several limitations in the assessment of scene-based tools, mainly due to *i)* human subjectivity in the data selection process, *ii)* dependency of results on external conditions (e.g., display and ambient light conditions), *iii)* impossibility of testing the technique on large amounts of heterogeneous data.

Cropping is a simple yet powerful way to maliciously alter the content and the meaning of an image, as shown in Fig. 1. Despite its communication impact, this kind of forgery has historically been less investigated by the forensic community than other image manipulations like splicing, copy-move or removal. Signal-based methods for cropping detection were proposed that look for block-

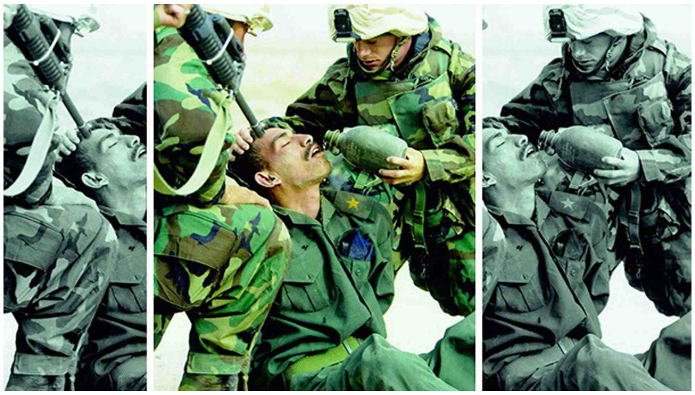


Figure 1: A famous example of ambiguity induced by image cropping. The original photo of an Iraqi soldier surrendered to the U.S. Army after crossing the border in Kuwait (centre) can be interpreted either in a threatening (left) or charitable (right) way by simply selecting which part of the image to remove. (Credit: AP Photo/Itsuo Inouye.)

ing artefacts arising from image compression [22, 23]. On the scene-based side, a semi-automatic approach to cropping detection based on the exploitation of vanishing points was recently proposed in [24].

Differently from our previous work [25], that mainly focuses on a reliability analysis of principal point estimation in a forensic scenario, conducted on synthetic data and validated manually on few real scenes, in this paper we introduce a new cropping detector that treasures the findings of [25]. Our detector works on single images of Manhattan-World scenes [26] (i.e. scenes of man-made environments that typically include buildings or structure shaving three main orthogonal directions) and exploits the camera principal point as scene-level trace. The main novelties of this new approach are twofold. On the one hand, it is fully automatic, and avoids by itself intractable images, thus improving detection performance. On the other hand, it is highly reliable, thanks to the introduction of robust estimation techniques and of a specially designed metric for the assessment of image integrity. The results of a comprehensive experimental campaign show the effectiveness of the approach.

The paper is organized as follows. The next Section provides the reader with a brief review of the computer vision techniques for principal point estimation and their adaptation to the forensic domain. Sec. 3 introduces theoretical and design issues underlying our detector—referred to AMARCORD (**A**utomatic **M**anhattan-scene **A**symmet**R**ically **C**r**O**pped image**R**y **D**etector). In particular, Subsec. 3.3 introduces and motivates AMARCORD's novel metric for robust cropping detection. Experimental results on several image datasets and scenarios are reported in Sec. 4. Finally, Sec. 5 summarizes the results achieved and highlights open issues.

2. The Principal Point and its use in Forensics

The principal point (PP) is an image point defined as the foot of the perpendicular from the camera centre to the image plane [27]. In pristine images, this point is very close to the image center, i.e., the point where the image diagonals meet. After any asymmetrical cropping manipulation, the image center moves to a new position determined by the new image dimensions while PP, being a camera-related parameter, remains still. AMARCORD leverages this invariance property of PP for detecting asymmetrical cropping based on the discrepancy between PP and the image centre.

PP estimation is a known topic in computer vision and photogrammetry, strictly related to the camera calibration problem. When the camera is available, accurate off-line techniques exploiting a known pattern in the scene can be used to calibrate it [28]. The calibration problem can also be solved in the absence of the original camera, provided that images taken with that camera are available, in which case the problem is better known as self-calibration. Several self-calibration techniques exist, which differ according to the type of visual data (videos, image collections, single images) and operating conditions (e.g., in a video, fixed vs changing camera parameters) [29]. Self-calibration of single images typically relies on a priori information about the scene structure, which can be exploited to infer the calibration parameters [30, 31]. Structural information of special relevance to applications is that of Manhattan-World scenes [26], where it is assumed that the scene includes man-made structures like buildings, giving rise to sets of lines having mutually orthogonal directions in 3D [32, 33]. These lines, once projected onto the image plane using a pinhole camera model, can be used to estimate the vanishing points (VPs) of the scene: Indeed, all the lines sharing the same 3D direction project onto a single VP in the image. For Manhattan-World scenes, which are composed of cube-like structures, most of the image lines are projections of three mutually orthogonal 3D directions, and calibration information—including PP—can be extracted from a triangle whose vertexes are the VPs related to those directions (see Fig. 2, and also [27, Ch. 8] for mathematical details).

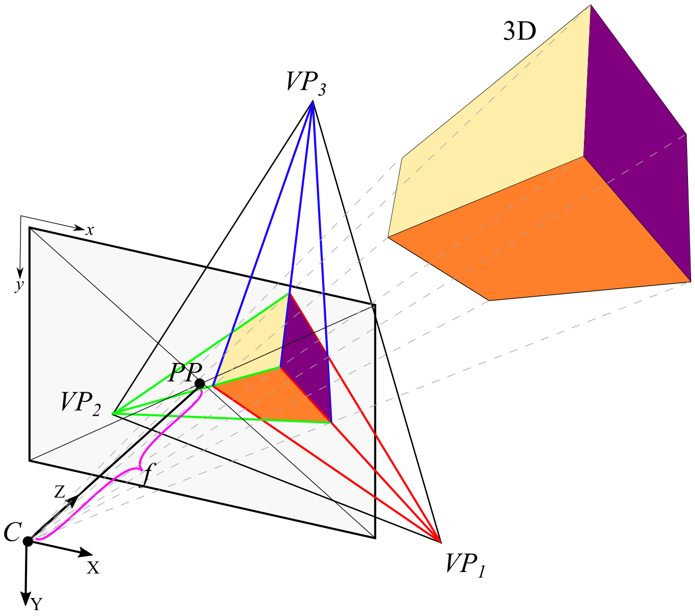


Figure 2: (Best viewed in color) Example of pin-hole projection of a cube-like object, from a camera center C. The 3D cube projected onto the image plane gives rise to an image where lines sharing the same 3D direction converge towards three vanishing points (red in VP_1, green in VP_2, and blue in VP_3). From these points the main camera parameters (i.e., the focal length f and the principal point PP) can be estimated.

Transferring to the forensic domain computer vision techniques, which typically assume genuine images, make the task of camera calibration (and specifically PP estimation) even more challenging. Indeed, in standard computer vision one is legitimate to use default settings to ease, improve and even avoid parameter estimation. For example, PP is often initially assumed to be in the image center, and then either used as is or slightly refined. Conversely, in common forensic scenarios, only images of unknown origin are available, and no a priori assumptions can be made about parameters, nor it is possible to rely on metadata (e.g., EXIF data), which could also have been manipulated. This means that any parameter to be exploited for tampering detection must be extracted directly from (possibly manipulated) image data, without any prior information about it.

Concerning PP, only a few published methods exist that try to exploit it as a clue for tampering detection. In [34] the authors presented a method based on the estimation of the homography that maps the eyes of a person onto the image plane. PP is then recovered by homography decomposition (supposing the focal length is known) and exploited for splicing detection. In [24], PP is estimated from three vanishing points related to mutually orthogonal directions using a set of manually selected image lines, and then exploited to detect cropping on Manhattan World scenes based on the Euclidean distance between PP and the image center. Slightly different, yet still related to this topic, is the approach described in [35], where the direct observation of vanishing points of buildings in the 3D scene is proposed as tampering detection feature in the place of PP.

3. Automatic Detection of Asymmetrical Cropping

AMARCORD is designed to detect evidence of cropping in a large collection of images. This requires that the algorithm must operate in an automatic way, being also capable to decide autonomously whether the image at hand is tractable (i.e., it meets the Manhattan-world scene assumption) or not.

After detection of straight lines (see Sect. 3.1), these are clustered in order to estimate a set of three vanishing points related to mutually orthogonal directions in 3D (see Sect. 3.2). Evidence of cropping is then established with a statistical analysis of a cloud of putative PPs extracted from the image (see Sect. 3.3). The heuristic criteria introduced to discard intractable images are discussed in Sect. 3.4. Robust computational techniques are employed throughout the algorithm, so as to cope with large sources of noise in the data and improve detection performance.

3.1. Line Segment Detection and Clustering

Line segments are obtained as a map of one-pixel thick edges by applying the Canny edge detector [36] followed by non-maxima suppression. Connected components are found using flood-fill, and split into straight edges based on the standard deviation of the fitted lines [37]. Fig. 3 shows an example of detected line segments superimposed to the image.

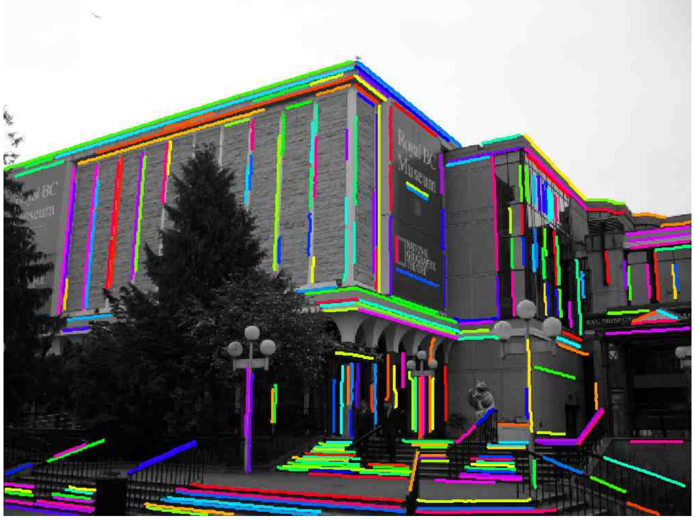


Figure 3: (Best viewed in color) An example of automatic line detection.

Given N detected image line segments, these are clustered according to the VPs they converge to. This is achieved through simultaneous estimation of multiple models with J-Linkage [38]. M initial VP candidates are determined as the intersection of two randomly selected line segments. A $N \times M$ preference matrix P is built, where $P_{i,j}$ is the preference score of the i-th edge for the j-th VP. $P_{i,j}$ is set to 1 if the distance between the i-th line segment and the j-h VP is below a consensus threshold, otherwise it is set to 0.

Under the assumption that edges converging to the same VP tend to have similar preference sets (i.e., rows of the preference matrix P), line segments are clustered by an iterative aggregation procedure based on the Jaccard distance [38]. The process ends when the distances between clusters are maximized, returning as output collections of lines converging to the same VP. Notice that J-Linkage can produce different outputs for a fixed given input, due to the random VPs initialization. To avoid such non-deterministic behaviour, the number of candidate VPs should exhaustively include all the $M = \frac{N(N-1)}{2}$ pairwise edge intersections, which is computationally infeasible for some images. However, experiments showed that setting $M = 5000$ gives a good compromise between computational and repeatability/accuracy requirements. In Fig. 4 an example of line clustering is reported, where in red, green and blue are shown the three most populated clusters, while other clusters are also reported with different colors.

3.2. Extraction of the VP triplet and estimation of PP

Based on the idea that in Manhattan scenes most of the lines usually belong to three dominant orthogonal directions (e.g., the sides of a building), AMARCORD chooses as VP candidates related to mutually orthogonal directions those originated from the most populated clusters returned by J-Linkage.

From each of the three selected clusters, a VP is obtained as the intersection of the cluster lines. Let $L_k = \{\mathbf{l}_i^k\}_{i=1,\ldots,I}$ be the set of all the lines in the k-th cluster, and $[a_i, b_i, c_i]$ be the parameters of $\mathbf{l}_i^k$—such that a general point $\mathbf{q} = (x_q, y_q)$ lies on $\mathbf{l}_i^k$ if and only if $a_i x_q + b_i y_q + c_i = 0$. All the cluster lines can be compactly represented by the matrix

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ & \vdots & \\ a_I & b_I & c_I \end{bmatrix} \tag{1}$$

The intersection point $\mathbf{v}_k$, i.e., the VP of the k-th cluster, can be obtained by solving the linear system $A\mathbf{v}_k = 0$ by least squares, where $\mathbf{v}_k$ is expressed in homogeneous coordinates. This first linear VP estimate can be then be refined

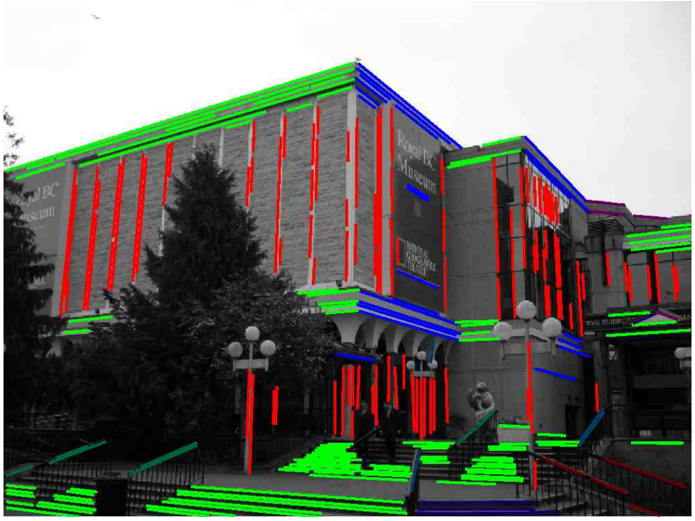


Figure 4: (Best viewed in color) Automatic line clustering for the image in Fig. 3. The three most populated clusters are shown in red, green and blue, respectively. Other less populated clusters are also reported with different colors (e.g., in dark-red, dark-blue, purple and dark-green).

by iterative non-linear optimization [27].

Notice that in practical scenarios the intersection of more than two concurrent lines inside a cluster is not unique, since noise can perturb line detection accuracy (see the detail of Fig. 5). In [25] we showed that well-spaced lines reduce the VP estimation error and that, on the other hand, employing many near-to-parallel lines does not improve on VP estimation, but only increases the computational time. Therefore, in AMARCORD we limit to $t = 20$ the maximum number of lines per cluster to be used for estimating each VP. To obtain a subset of well spaced lines, we adopt the following *line selection scheme*. First, we compute the "vanishing angle" [25], i.e., the maximum possible angle among those obtained by intersecting pairwise all the lines in the cluster, and split it into $t-1$ angular sectors. Then, for each sector we select the line that is closest to the bisector.

Once the three mutually orthogonal VPs are obtained, PP is estimated by solving a linear system [27]. Explicitly, each pair of VPs, $(\mathbf{v}_i, \mathbf{v}_j)$, with $i \neq j$,

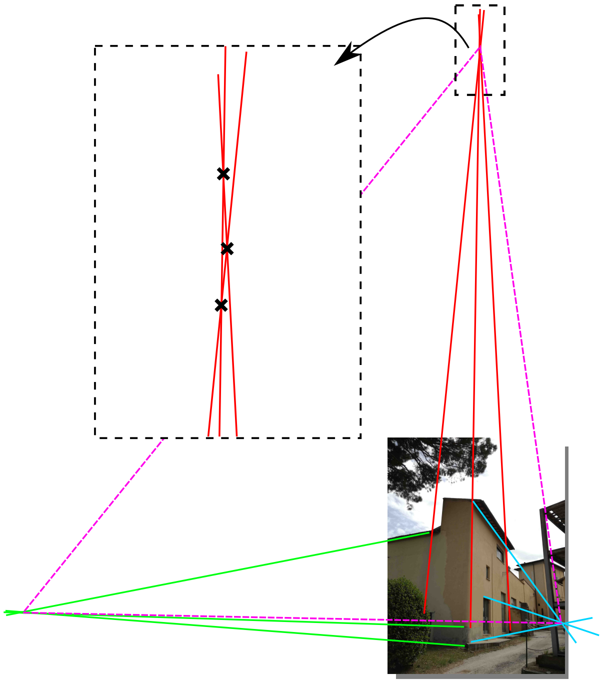


Figure 5: (Best viewed in colour) Estimation of vanishing points: As shown in the magnified area—where the ✖s indicate the intersection points—line intersection is not unique due to noise.

defines a constraint

$$(K^{-1}\mathbf{v}_i)^\top(K^{-1}\mathbf{v}_j) = \mathbf{v}_i^\top(KK^T)^{-1}\mathbf{v}_j = 0 \tag{2}$$

where K is a camera calibration matrix with three degrees of freedom (focal length, plus the two coordinates of PP). The three VPs suffice to estimate $(KK^T)^{-1}$ and eventually K (hence PP) by Cholesky factorization.

3.3. Cropping detection based on a statistics-aware metric

After PP estimation, a simple way to decide whether the image was cropped or not is to evaluate the normalized Euclidean distance

$$\mathcal{D}_2(\mathbf{p}, \mathbf{c}) = \frac{\| \mathbf{p} - \mathbf{c} \|}{\sqrt{w^2 + h^2}} \tag{3}$$

between the principal point **p** and the camera center **c**, where normalization is done w.r.t. the diagonal of a $w \times h$ image [24, 25]. The larger is the distance, the more probable is that a cropping event has occurred.

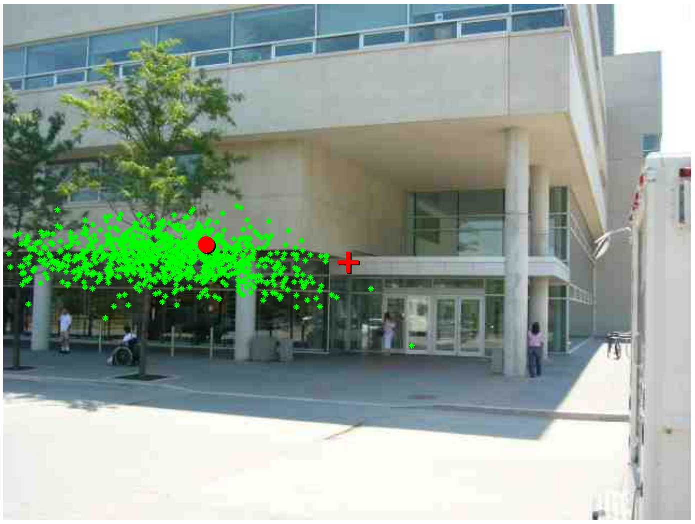


Figure 6: (Best viewed in colour) A pristine image. The ground truth PP is indicated by a red cross, while the "one-shot" estimated PP is shown as a red dot. The green point cloud shows the PPs obtained after 1000 Monte Carlo iterations (see text).

However, the above procedure is extremely sensitive to noise in the measurements, as it relies on a single, "one-shot" estimate of PP from the content (i.e., length, orientation and distribution of the lines) of the image at hand. An explanation of this fact is given with the help of Fig. 6. The figure shows a pristine (i.e., not cropped) image, in which the vertical VP is quite difficult to estimate, since the corresponding image lines are almost parallel to each other. As a result, the PP estimated as explained in Subsec. 3.2 (indicated by a red dot) is located quite far from the ground truth PP (i.e. the image center, indicated by a red cross), although in a noise-free scenario these two points would be coincident.

In order to gain an insight into uncertainty in line detection, for each VP cluster, the lines (also estimated from the image as explained in Subsec. 3.2) were repeatedly perturbed by adding a zero-mean Gaussian noise with a standard deviation of $\sigma = 0.3$ pixels to one of the line ends. The green dots of Fig. 6 indicate the positions of the PPs resulting after 1000 line perturbation iterations. This Monte Carlo simulation confirms the fact that horizontal uncertainty in PP estimation is very large. Indeed, the cloud is highly scattered along the horizontal direction, with some PPs located very far away from the ground truth, and others quite near to it, hence closer to the true solution than the PP estimated "one-shot".

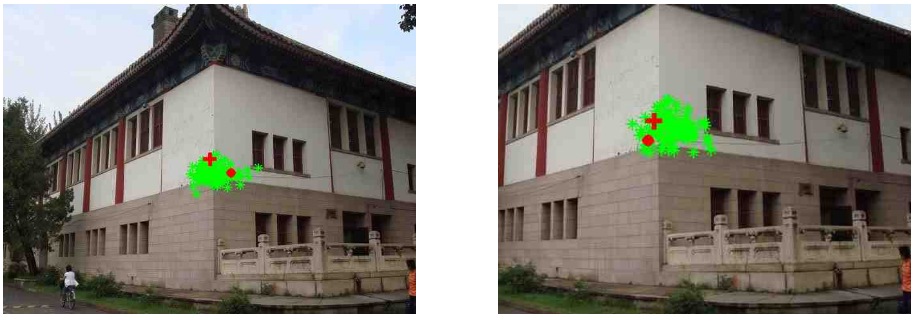
Figure 7 shows the PP clouds obtained before and after image cropping. The clouds have similar shapes and remain almost fixed w.r.t. the image content. Notice that, although both clouds contain the ground truth solution, in neither case the "one-shot" solution coincides with it.

The above observations inspired us to introduce a new metric for the cropping problem, which is based on a whole cloud of PPs obtained by a Monte Carlo process similar to the one described above. Unlike the Euclidean distance, which implicitly assumes that PP is a deterministic variable, our metric regards each of the cloud points as a sample of the statistical distribution of PP, considered here as a random variable. The new metric, referred to as $\mathcal{D}_{p\%}$, is computed in two steps. First, the distribution of the Euclidean distance between PP and the image center is estimated by using all the PPs in the cloud. Then, $\mathcal{D}_{p\%}$ is estimated as the value corresponding to the p-th percentile of the distance distribution. The best percentile value p was obtained experimentally, as reported in Sect. 4.3.

3.4. Opt-out criteria for improved reliability

AMARCORD is deemed to give erroneous results on images characterized by lack of detectable lines or, on the opposite, an excessive number of lines not belonging to mutually orthogonal directions. Other critical images are those exhibiting extreme viewpoints (with one or more VPs going to infinity) or depicting non-Manhattan scenes.

In order to automatically detect the above conditions, we introduced the following two heuristic criteria to check the status of AMARCORD at runtime.



(a)

(b)

Figure 7: (Best viewed in colour) Example of PP clouds obtained through Monte Carlo simulations: (a) a pristine image and (b) its 20% cropped version. Red crosses indicate the ground truth PP, while red dots indicate the "one-shot" PP. Note that the two clouds have similar shapes and have a stable position w.r.t. the image content.

If either of these criteria is not satisfied, the analysis is aborted and the input image is labelled as *intractable.* This simultaneously reduces the computation time by avoiding to analyze in detail inappropriate image content while processing a large collection of casual images, and helps increasing cropping detection reliability by reducing false alarms.

Max Angle. As reported in [39], a triangle joining vanishing points related to three mutually orthogonal directions in the 3D space can't have angles wider than 90°. If AMARCORD finds such a configuration for the computed VPs, the image is discarded immediately, without wasting additional time on its analysis.

Max Dist. As shown in the Appendix, the distance between the ground truth PP and the cropped image centre (normalized w.r.t. the diagonal of the cropped image) can be expressed as a function of the cropping factor $\alpha \in [0, 1[$ as

$$\mathcal{S}(\alpha) = \frac{\alpha}{2(1-\alpha)} \tag{4}$$

Since AMARCORD is assumed to handle cropping factors up to 50%, with maximum expected distance equal to $\mathcal{S}(1/2) = 0.5$, the image at hand is discarded without entering the Monte Carlo analysis if the "one-shot" distance $\mathcal{D}_2(\mathbf{p}, \mathbf{c})$ of Eq. 3 exceeds 0.5.

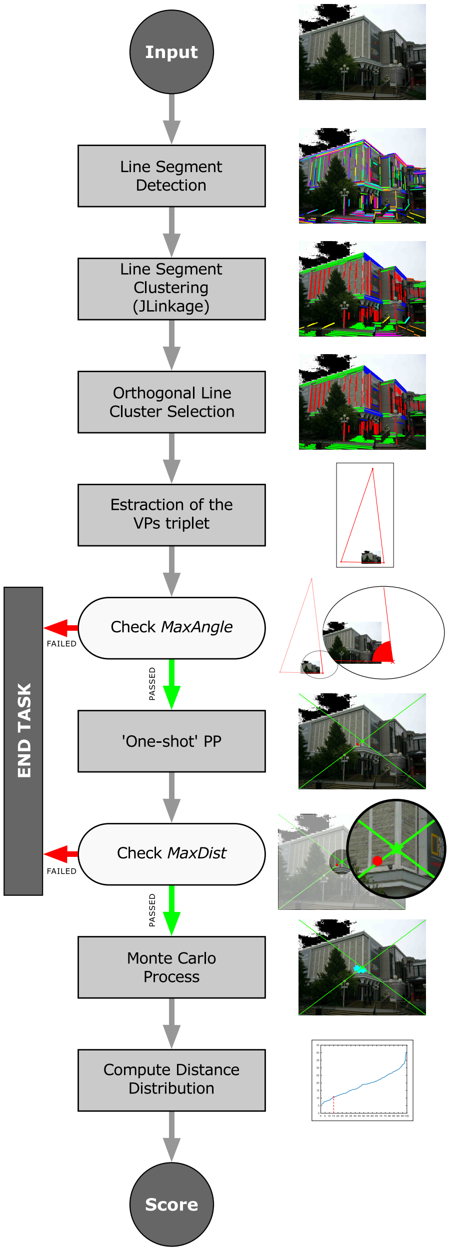

Input
Line Segment Detection
Line Segment Clustering (JLinkage)
Orthogonal Line Cluster Selection
Estraction of the VPs triplet
Check MaxAngle
FAILED
PASSED
'One-shot' PP
Check MaxDist
FAILED
PASSED
END TASK
Monte Carlo Process
Compute Distance Distribution
Score

Figure 8: (Best viewed in colour) The AMARCORD block diagram, including heuristic opt-out checking. If the *MaxAngle* or *MaxDist* criteria are not satisfied, AMARCORD stops the analysis and saves time for the next image.

Algorithm 1 AMARCORD

```
procedure AMARCORD($I$)                                      ▷ $I$: input image.
    $cc \leftarrow [I_w/2, I_h/2]$                           ▷ $cc$: image center
    $L \leftarrow DetectLines(I)$                            ▷ See Sect. 3.1
    $K \leftarrow JLinkageClustering(L)$
    if $Size(K) \geq 3$ then
        $K_H \leftarrow SelectOrthoClusters(K)$
        $[vp_0, vp_1, vp_2] \leftarrow ComputeVPs(L, K_H)$   ▷ See Sect. 3.2
        if $MaxAngle(vp_0, vp_1, vp_2) < th_{MA}$ then      ▷ See Sect. 3.4
            $pp \leftarrow OneShotPP(vp_0, vp_1, vp_2)$
            if $MaxDist(pp, cc) \leq 0.5$ then               ▷ See Sect. 3.4
                $pp_{MC} \leftarrow getPPbyMonteCarlo(L, K_H)$   ▷ See Sect. 3.3
                $dDist \leftarrow getDistanceDitribution(pp_{MC}, cc)$
                return $\mathcal{D}_{p\%}$
            else
                return $-1$                                  ▷ Fail $MaxDist$
        else
            return $-1$                                      ▷ Fail $MaxAngle$
    else
        return $-1$
```

In Figure 8 the block diagram of the whole AMARCORD framework is shown, complete with opt-out checking. We also report a pseudo-code version of AMARCORD in Alg. 1.

4. Experimental Results

4.1. Datasets

AMARCORD was tested on four datasets of images depicting Manhattan-World scenes: the York Urban Line Segment database [40] (YDB), the PKU Campus Database [41] (PKU), the Toulouse Vanishing Points dataset [42] (TVPD), and our new Florence Building dataset (FLB). YDB includes 102 images of urban environments captured inside the campus of York University and in downtown Toronto, Canada. The images are taken with a calibrated Panasonic Lumix DMC-LC80 digital camera. The database provides also camera calibration parameters and ground-truth line segments. PKU includes 200 photos of Manhattan-World scenes with VP ground-truth. However, we noticed that for

45 images one of the three orthogonal VPs is missing in the ground-truth: We removed those images from the evaluation, thus reducing PKU to 155 photos, in order to present a fair comparison of results obtained with and without ground-truth information. TVPD contains 114 images of Manhattan scenes taken with an iPad Air 1, with associated inertial measurement and VP ground-truth with uncertainty regions. Note that all the man-made datasets available online (YDB, PKU, and TVPD) do not specify whether the images are camera-native or have been previously processed. In order to make a fair comparison between AMARCORD and the signal-based method of [22] (see Sect. 4.9) we created a new dataset of man-made scenes, named Florence Buildings (FLB). Using a Canon 5D Mark II camera, we captured 94 raw images of man-made environments, with a resolution of 5616x3744 pixels, and then we saved them with jpeg compression using two quality factors: 50 and 90. Note that we do not provide ground-truth lines for this new dataset. Finally, we built a Natural dataset (NAT), composed by scenes not satisfying the Manhattan-World assumption, to test the system capability to exclude intractable input; these images were gathered from the VISION dataset [43] by manually selecting scenes without man-made structures.

For the purpose of experimental data generation, we developed a MATLAB script working on the dataset images. The script builds $3 \times 4 = 12$ different cropping sets for each dataset, by varying the cropping percentage—$20\%, 35\%$ and 50%—and the cut orientation—upper-left (UL), upper-right (UR), bottom-left (BL), and bottom-right (BR). Additionally, ground-truth line segments (when available) were modified according to the cut, in order to remove or shorten lines falling out of the cropped image area. Notice that image aspect ratio was preserved by cutting along two consecutive sides with the same percentage. Evaluating different cropping percentages is motivated by the fact that wider cropping should theoretically be easier to detect as PP is farther from the image centre, but in practical situations strongly cropped images present less visible edges to be detected, thus making it more difficult to apply AMARCORD.

Table 1 summarizes the characteristics of each dataset.

Table 1: The list of datasets used to test AMARCORD. For each dataset we report its name, the number of images (#IMG), if ground-truth lines are available (w/GT), and the kind of scene depicted.

Name	#IMG	w/GT	Scene
York Urban Line Segment DB (YDB)	102	Y	Man-made
PKU Campus Database (PKU)	155	Y	Man-made
Toulouse Vanishing Points DS (TVPD)	114	Y	Man-made
Florence Buildings DB (FLB)	94	N	Man-made
Natural (NAT)	110	N	Natural

In the following subsections, the proposed pipeline was evaluated according to the Receiver Operating Characteristic (ROC) curve, considering true positive rate (TPR) versus false positive rate (FPR). The area under the ROC curve (AUC) was also used as a global assessment index.

In order to improve readability of this section, in Table 2 we summarize the different experimental setups, reporting for each test a brief description, the used data, and other relevant information.

Table 2: Experimental setup summary. For each test we report its Section, a brief description, the used data, if GT lines or automatically detected lines are used, counter-forensics method used and if opt-out criteria are considered.

Sect.	Description	Dataset	GT	Detection	Counter-forensics	Opt-out
4.2	Evaluate effectiveness of the proposed *line selection scheme* for VP computation (see Sect. 3.2)	YDB	Y	N	None	N
4.3	Select the best p-value for $\mathcal{D}_{p\%}$ (see Sect. 3.3)	YDB, PKU, TVPD	N	Y	None	N
4.4	Full AMARCORD evaluation	YDB, PKU, TVPD, NAT	Y	Y	None	N
4.5	Assess J-Linkage clustering performance	YDB	Y	N	None	N
4.6	Evaluate AMARCORD robustness *vs* recompression through social network	YDB, PKU, TVPD	N	Y	Recompression	N
4.7	Evaluate AMARCORD robustness *vs* enhancement	YDB, PKU, TVPD	N	Y	Equalization of lighting channel	N
4.8	Evaluate AMARCORD robustness *vs* filtering	YDB, PKU, TVPD	N	Y	Gaussian smoothing	N
4.9	Comparison *vs* signal-based method [22]	FLB	N	Y	None, Recompression	N
4.10	Evaluate opt-out criteria	YDB, PKU, TVPD, NAT	N	Y	None	Y

4.2. Efficient estimation of vanishing points

Table 3 reports AUC values obtained for each cropping set on YDB, with and without the computational optimization proposed in Sec. 3.2. Tests are reported only using groud-truth (GT) lines and simple normalized 'one-shot' Euclidean distance, to prevent other pipeline intermediate factors, such as the reliability of the Canny edge detector, to affect the output.

Table 3: AUC for the proposed pipeline without and with the line selection scheme.

Without/**With**	20%	35%	50%
UL	0.8157 / **0.8018**	0.9204 / **0.9261**	0.9771 / **0.9812**
UR	0.8223 / **0.8105**	0.9244 / **0.9387**	0.9741 / **0.9829**
BL	0.7761 / **0.7985**	0.9186 / **0.9271**	0.9788 / **0.9840**
BR	0.7935 / **0.8014**	0.9119 / **0.9398**	0.9748 / **0.9825**

Results clearly show no relevant drop in the detector reliability. Yet, well spaced lines tends to reduce the PP error estimation, slightly improving AUC. Additionally, running times are halved: on a Ubuntu 16.04 workstation mounting an Intel Core2 Q9400 @ 2.66 GHz and 8 GB RAM the average time spent on an image decreases from 59s to 30s.

4.3. Selection of percentile for $\mathcal{D}_{p\%}$

To select the best percentile p for the novel metric introduced in Sec. 3.3 we run AMARCORD on YDB, PKU, and TVPD with all crop percentages, varying $p \in [5, 50]$, with steps of 5. Results are scored with the obtained AUC. Note that we do not use any of the criteria introduced in Sect. 3.4. In Table 4 we report all the AUCs obtained.

Table 4: Percentile analysis: for each man-made dataset, and for each cutting percentage, we report the AUCs obtained by $\mathcal{D}_2$ and $\mathcal{D}_{p\%}$ respectively, considering $p \in [5, 50]$. In **bold** the best AUCs. Notice that $\mathcal{D}_{p\%}$ performs better than $\mathcal{D}_2$ regardless of the p value.

Dataset	$\mathcal{D}_2$ **AUCs**	$\mathcal{D}_{p\%}$ **AUCs**									
		5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
YDB 20%	0.5905	0.6445	0.6494	**0.6518**	0.6495	0.6474	0.6435	0.6390	0.6354	0.6319	0.6265
YDB 35%	0.7266	0.7521	0.7575	0.7570	0.7585	0.7586	**0.7601**	0.7591	0.7598	0.7585	0.7573
YDB 50%	0.8378	0.8443	0.8463	0.8499	0.8521	0.8523	0.8532	0.8532	**0.8535**	0.8524	0.8520
YDB Avg	0.7183	0.7470	0.7511	0.7529	**0.7533**	0.7528	0.7523	0.7504	0.7496	0.7476	0.7453
PKU 20%	0.5867	**0.5938**	0.5930	0.5922	0.5925	**0.5938**	0.5925	0.5919	0.5921	0.5921	0.5919
PKU 35%	0.6812	0.7039	**0.7047**	0.7040	0.7034	0.7035	0.7024	0.7018	0.7015	0.7015	0.7000
PKU 50%	0.7389	0.7621	0.7666	**0.7673**	0.7669	0.7671	0.7662	0.7669	0.7669	0.7664	0.7661
PKU Avg	0.6689	0.6866	**0.6881**	0.6879	0.6876	**0.6881**	0.6870	0.6868	0.6868	0.6867	0.6860
TVPD 20%	0.6497	0.6728	0.6737	**0.6747**	0.6727	0.6711	0.6690	0.6679	0.6653	0.6617	0.6620
TVPD 35%	0.7665	**0.7856**	0.7845	0.7827	0.7798	0.7802	0.7803	0.7798	0.7804	0.7788	0.7777
TVPD 50%	0.8348	0.8410	0.8434	0.8439	0.8455	0.8460	0.8460	**0.8463**	0.8460	0.8453	0.8443
TVPD Avg	0.7504	0.7664	**0.7672**	0.7671	0.7660	0.7658	0.7651	0.7647	0.7639	0.7619	0.7613
All 20%	0.6090	0.6370	0.6387	**0.6396**	0.6382	0.6374	0.6350	0.6329	0.6309	0.6286	0.6268
All 35%	0.7248	0.7472	**0.7489**	0.7479	0.7472	0.7474	0.7476	0.7469	0.7472	0.7462	0.7450
All 50%	0.8038	0.8158	0.8188	0.8204	0.8215	0.8218	0.8218	**0.8221**	**0.8221**	0.8214	0.8208
All Avg	0.7125	0.7333	0.7355	**0.7359**	0.7356	0.7356	0.7348	0.7340	0.7334	0.7321	0.7309

As can be noticed, the new metric $\mathcal{D}_{p\%}$ obtain higher AUCs for every value of p w.r.t. $\mathcal{D}_2$: this proves the effectiveness of the proposed solution. To select the best p value, we consider the average AUC results obtained on all the datasets and on all the cropping percentages (reported in the last row of Tab. 4), and we look for the maximum AUC: best results are obtained setting $p = 15\%$ achieving an AUC of 0.7359, considering all the dataset and all the cuts.

4.4. Cropping detection results

Hereafter we report the results obtained on YDB, PKU, TVPD, and NAT datasets. For each test we report performances as ROC and AUC obtained using both metrics: $\mathcal{D}_2$ and $\mathcal{D}_{15\%}$. Note that AMARCORD with the $\mathcal{D}_2$ metric can be seen as a fully automatic version of the semi-automatic solution proposed in [24]. Only for the YDB, PKU, and TVPD datasets we report also results obtained using the ground-truth lines and clustering, since those information are obviously missing for the NAT dataset. Notice also that, for the sake of brevity, we present here only results for upper-left asymmetric crops, since we observed that different cutting orientations produce very similar results. These results are obtained by setting to infinity both the opt-out thresholds of Sect. 3.4 (see Sect. 4.10 for the results obtained by introducing opt-out criteria).

In table 5 we report AUC values obtained with the different setups (metric/cropping percentage) on the four datasets. Then, Fig. 9 presents ROC curves for YDB. Similarly, Fig. 10 and Fig. 11 show ROCs respectively for PKU and TVPD. Finally, Fig. 12 reports ROCs for the NAT dataset.

As can be observed, all the Manhattan-world datasets present similar performance—with PKU showing slightly inferior AUCs. Also, the newly introduced metric $\mathcal{D}_{15\%}$ always obtains higher AUC w.r.t. the more classical $\mathcal{D}_2$ and, as expected, using ground-truth information, results generally improve (+0.15 in average). This suggests that the main criticism of the AMARCORD pipeline is in the extraction and clustering of image lines: For this reason in the next section (Sect. 4.5), we will present tests aimed at better assessing the main source of errors.

Finally, results on natural images (NAT) are close to random guess, since these input are not tractable by AMARCORD. As described in Sect. 3.4, some heuristics can be defined to let the system discard unwanted inputs. In Sect. 4.10, the effectiveness of such criteria is demonstrated experimentally.

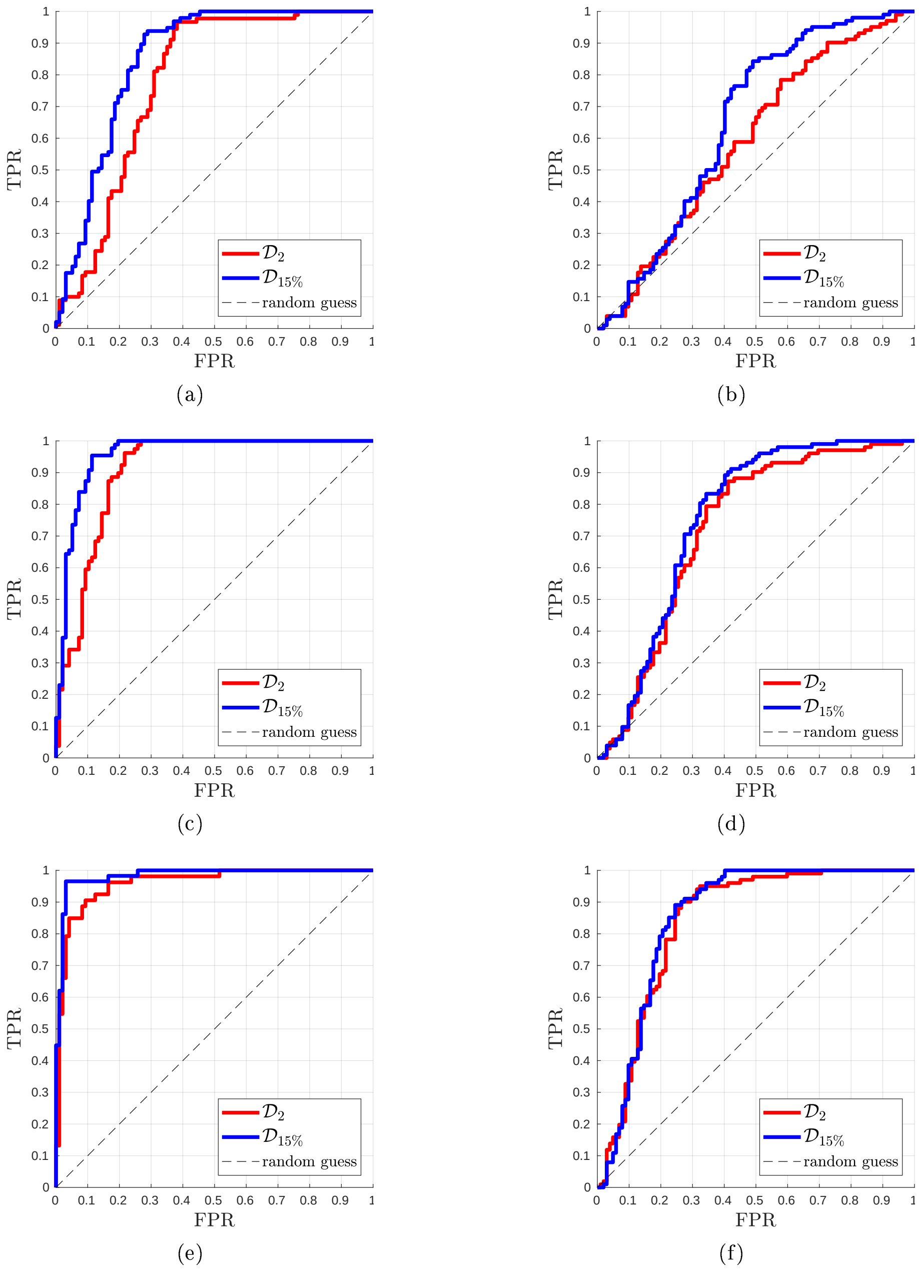
Table 5: Cropping detection AUCs on the four datasets for three different cropping percentage, using the $\mathcal{D}_2$ and $\mathcal{D}_{15\%}$. Top rows show results obtained using ground-truth lines and vanishing point clustering; Bottom rows present results of the fully automatic pipeline. Note that for the NAT dataset ground-truth line are missing.

	Metric	Crop%	YDB	PKU	TVPD	NAT
with GT	$\mathcal{D}_2$	20%	0.7786	0.6625	0.8498	—
		35%	0.9058	0.7902	0.9227	—
		50%	0.9593	0.8720	0.9541	—
	$\mathcal{D}_{15\%}$	20%	0.8506	0.7069	0.8704	—
		35%	0.9557	0.8319	0.9302	—
		50%	0.9828	0.8959	0.9643	—
without GT	$\mathcal{D}_2$	20%	0.5905	0.5867	0.6497	0.5412
		35%	0.7266	0.6812	0.7665	0.5641
		50%	0.8378	0.7389	0.8348	0.5603
	$\mathcal{D}_{15\%}$	20%	0.6518	0.5923	0.6747	0.5421
		35%	0.7570	0.7040	0.7827	0.5534
		50%	0.8499	0.7673	0.8439	0.5511

4.5. Results using GT lines without association to VPs

In this test we use the ground-truth lines as input, while not using the information regarding their VPs association. In this way, we can better assess J-Linkage line clustering performance independently from the line detection algorithm. In Table 6 AUCs obtained for YDB are shown, while we report in the additional material all the related ROC plots. As can be noticed, AUC values are very close to those reported in the upper part of Tab. 5 (using ground-truth lines) for YDB, since performance decreases by only 0.0101 in average.

However, note that the ground-truth lines include only segments belonging to one of the three dominant orthogonal directions, and no distractor lines are present, while in the fully automatic approach distractor lines—related to other 3D directions—are also present. In our opinion, the most critical aspect of the automatic pipeline is indeed the inclusion of noisy line segments into the

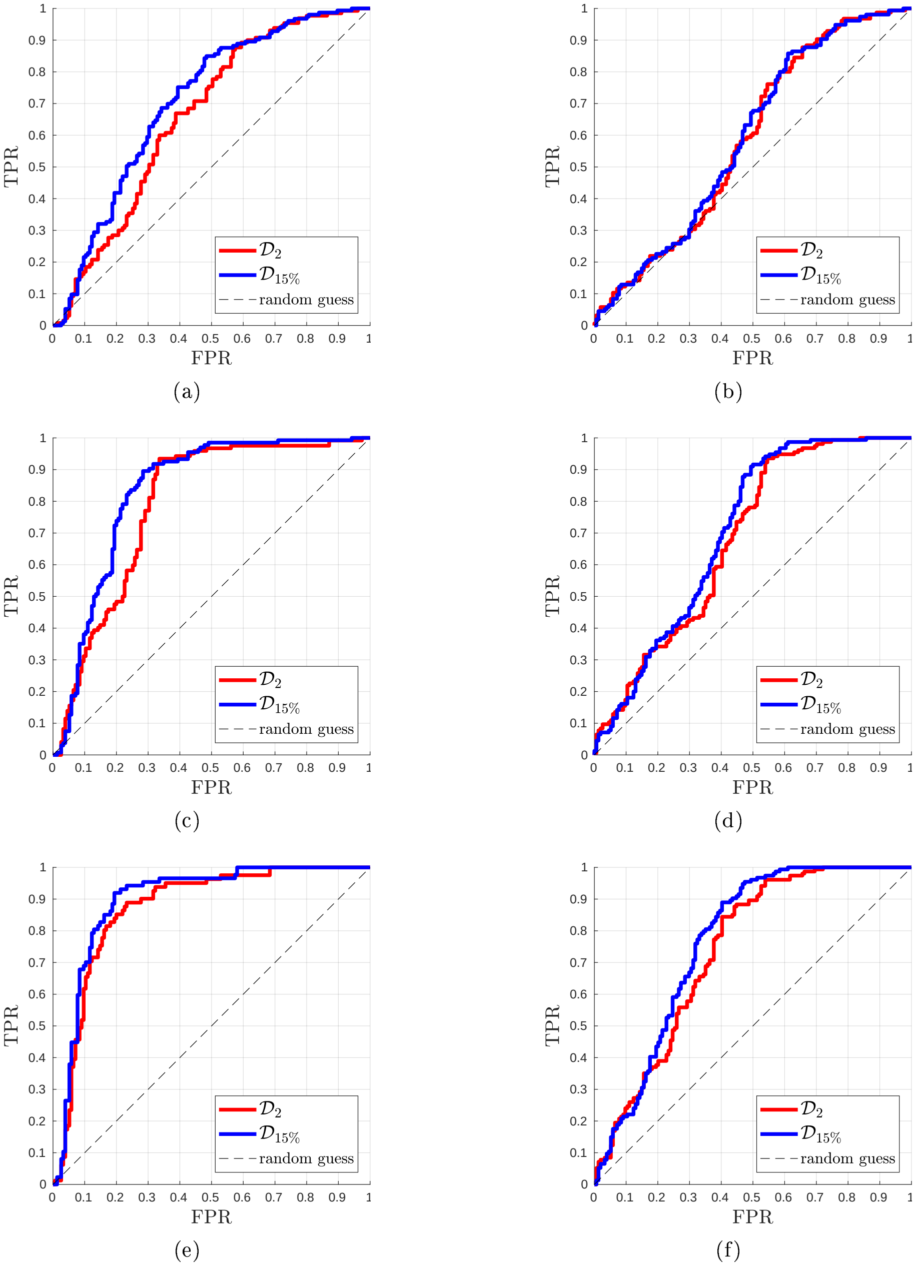


(a) (b)

(c) (d)

(e) (f)

Figure 9: (Best viewed in color) ROCs obtained from YDB with GT lines (left) and automatic line detection (right), respectively for 20% (a), (b), 35% (c), (d) and 50% (e), (f) crops.



(a)

(b)

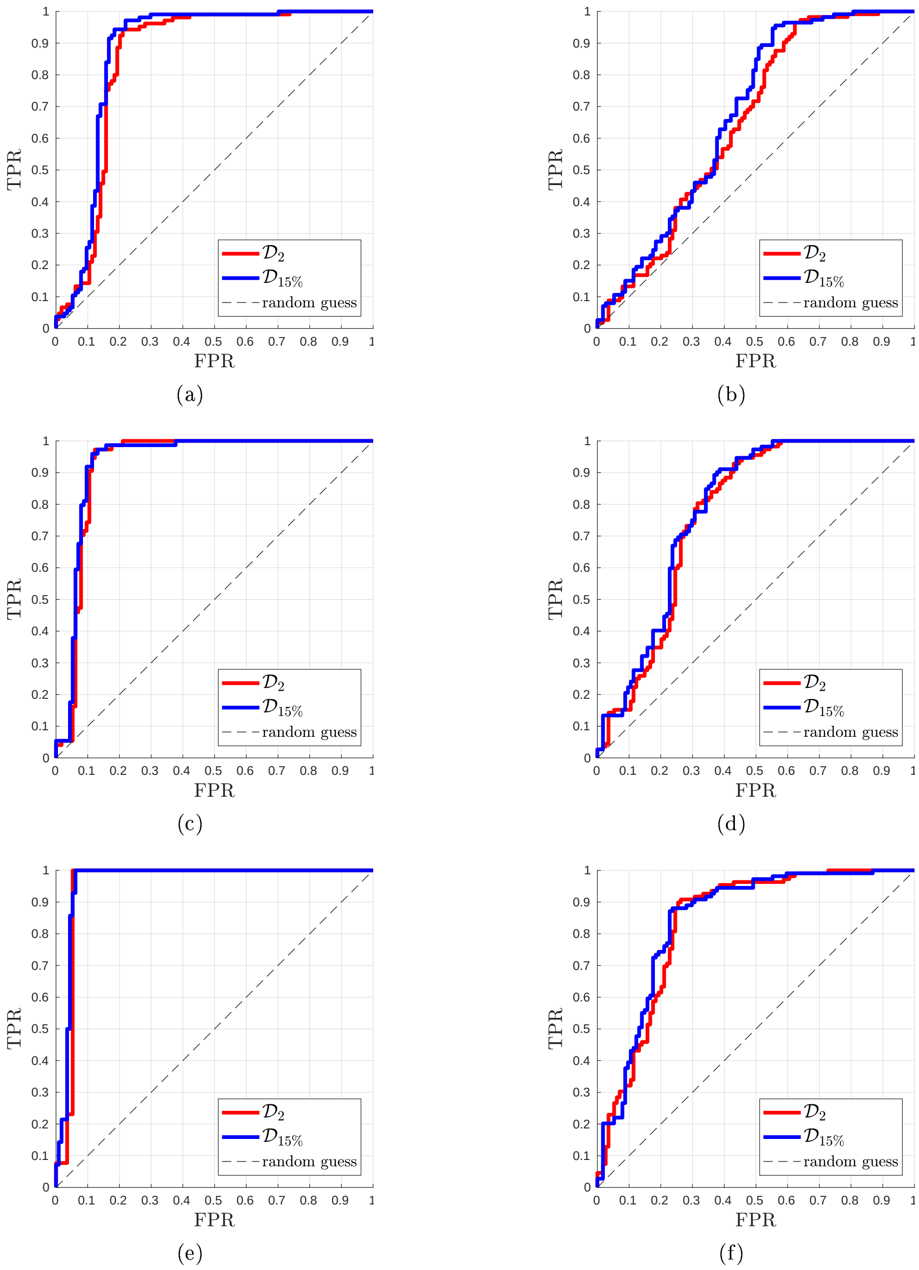
(c)

(d)

(e)

(f)

Figure 10: PKU DB with GT lines (left) and automatic line detection (right), respectively for 20% (a), (b), 35% (c), (d) and 50% (e), (f) crops.



(a)

(b)

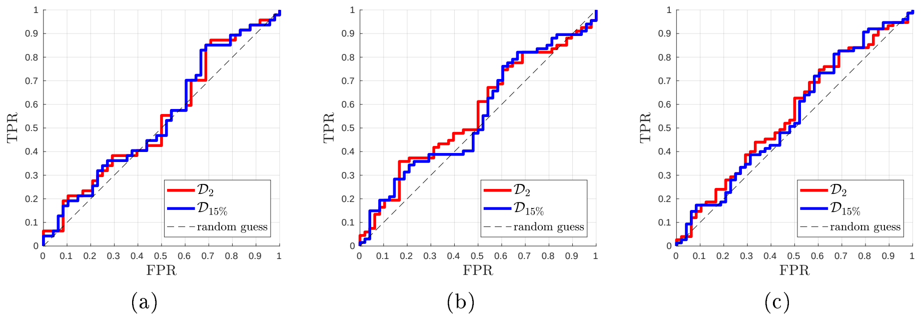
(c)

(d)

(e)

(f)

Figure 11: TVPD DB with GT lines (left) and automatic line detection (right), respectively for 20% (a), (b), 35% (c), (d) and 50% (e), (f) crops.



(a) (b) (c)

Figure 12: Natural DB with automatic line detection, respectively for 20% (a), 35% (b) and, 50% (c) crops.

analysis: In future work we will address this issue more deeply, trying to devise a learning-based method to discard distractor lines.

Table 6: Detection results on YDB, using ground-truth lines without the related VPs association. To help the comparison, we reports again AUC obtained with the full GT, already presented in Tab. 5.

Metric	**Crop%**	**AUC**	**AUC with full GT**
$\mathcal{D}_2$	20%	0.7401	0.7786
	35%	0.8818	0.9058
	50%	0.9468	0.9593
$\mathcal{D}_{15\%}$	20%	0.8234	0.8506
	35%	0.9501	0.9557
	50%	0.9875	0.9828

4.6. Results on recompressed images

In order to evaluate the robustness of our method against counter-forensics approach such as recompression—that for example could spoils the blocking-artefacts traces used by signal-based methods (e.g. [22, 23], see also Sect. 4.9)—all the dataset images have been uploaded and downloaded from Facebook, thus being recompressed automatically by the social network.

Results on recompressed images are reported in Table 7 while ROC plots are shown in the additional material. Note that, within the pipeline, the automatic line detection is the only step that can be strongly affected by image recompression. So we limit this test on the fully automatic pipeline, without considering ground-truth lines.

Table 7: Cropping detection AUCs after image recompression through Facebook. For each AUC we show in parentheses the difference w.r.t. the results reported in Tab. 5) for non-recompressed images. Note that only slight AUC reductions are measured, with a maximum decrease of 0.04.

Metric	Crop%	YDB		PKU		TVPD	
$\mathcal{D}_2$	20%	0.5801	(−0.0104)	0.5430	(−0.0437)	0.6404	(−0.0093)
	35%	0.7087	(−0.0179)	0.6647	(−0.0165)	0.7589	(−0.0076)
	50%	0.8145	(−0.0233)	0.7276	(−0.0113)	0.8376	(+0.0028)
$\mathcal{D}_{15\%}$	20%	0.6339	(−0.0179)	0.5505	(−0.0418)	0.6626	(−0.0121)
	35%	0.7386	(−0.0184)	0.6779	(−0.0261)	0.7724	(−0.0103)
	50%	0.8396	(−0.0103)	0.7439	(−0.0234)	0.8519	(+0.0080)

Comparing this results with those reported in the lower part of Tab. 5, it is evident that our method can handle effortlessly recompressed images without any particular drawback in performance, that decrease by only 0.02 in average. This is due to AMARCORD ability to exploit physical elements of the scene (i.e. line segments), which are more robust against counter-forensics attacks.

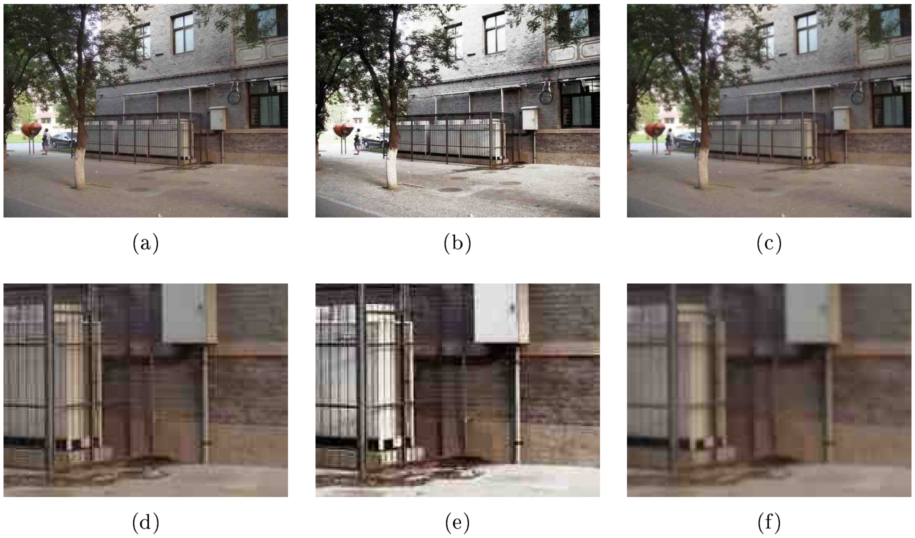


Figure 13: *Above*: Examples of enhanced and filtered images. (a) original image. (b) the same image after equalization of the lighting channel. (c) probe after Gaussian smoothing. *Below*: Zoomed-in versions to better inspect image changes due to enhancing (e) and filtering (f).

4.7. Results on enhanced images

In this Section we present results on a different counter-forensic attack: image enhancement. In particular, we chose to apply on each image of the YDB, PKU, and TVPD datasets (and on the relative cropped probes) an equalization of the lighting channel: firstly the image is mapped from the RGB to the HSL color space, then the L-channel is equalised, and finally the image is reported back to the RGB space. It can be seen by comparing Fig. 13(a) and 13(b) (and in their zoomed version in Fig. 13(d) and 13(e)), that the the transformation above causes strong variations to the image appearance, with different colors and slightly sharper edges. Enhanced probes are then saved as jpeg with a quality factor of 100. Table 8 reports the obtained AUCs (ROC plots for each datasets can be found in the additional material). It can be noticed that results after image enhancement are very close to those reported in Sect. 4.4 with native images: Except for two cases on YDB, performance drops only slightly—in average a reduction of 0.03 on the AUC. Note also that the $\mathcal{D}_{15\%}$ metric still show better results than $\mathcal{D}_2$.

Table 8: Cropping detection AUCs after image enhancement by lighting channel equalization. For each AUC we show in parentheses the difference w.r.t. the results reported in Tab. 5). Performance is quite stable, with a difference of at most 0.07.

Metric	**Crop%**	**YDB**	**PKU**	**TVPD**
	20%	0.6104 (+0.0199)	0.5138 (−0.0729)	0.6089 (−0.0408)
$\mathcal{D}_2$	35%	0.7156 (−0.0110)	0.6527 (−0.0285)	0.7354 (−0.0311)
	50%	0.8353 (−0.0025)	0.7214 (−0.0175)	0.8099 (−0.0249)
	20%	0.6185 (−0.0333)	0.5454 (−0.0469)	0.6265 (−0.0482)
$\mathcal{D}_{15\%}$	35%	0.7508 (−0.0062)	0.6759 (−0.0281)	0.7506 (−0.0321)
	50%	0.8579 (+0.0080)	0.7315 (−0.0358)	0.8314 (−0.0125)

4.8. Results on filtered images

Differently from the previous Section, here we test AMARCORD robustness against filtering effects: in particular we apply on all the probe images a Gaussian smoothing with $\sigma = 1$ over a square window of 5×5. As shown in Fig. 13(c) (and in its zoomed version Fig. 13(f)), applying this transformation results in blurred images with soft edges, as with the defocus effect. AUC results are reported in Table 9, while ROC plots are attached in the additional material. Even in this case AMARCORD shows stable results, insensitive to the filtering operation: indeed only slight reduction in AUC are found, with an average reduction of 0.02.

Table 9: Cropping detection AUCs after after image filtering by Gaussian smoothing. For each AUC we show in parentheses the difference w.r.t. the results reported in Tab. 5) for non-filtered images. Even in this case, only slight AUC drops are observed.

Metric	**Crop%**	**YDB**	**PKU**	**TVPD**
	20%	0.5793 (−0.0112)	0.5669 (−0.0198)	0.6340 (−0.0157)
$\mathcal{D}_2$	35%	0.7000 (−0.0266)	0.6822 (+0.0010)	0.7740 (+0.0075)
	50%	0.7816 (−0.0562)	0.7448 (+0.0059)	0.8390 (+0.0042)
	20%	0.6141 (−0.0377)	0.5728 (−0.0195)	0.6481 (−0.0266)
$\mathcal{D}_{15\%}$	35%	0.7312 (−0.0258)	0.6846 (−0.0194)	0.7824 (−0.0003)
	50%	0.8107 (−0.0392)	0.7330 (−0.0343)	0.8370 (+0.0069)

4.9. Comparison vs *signal-based cropping detector*

With this test we aim at comparing AMARCORD against the cropping detector presented in [22]. Differently from our solution, Bruna et al. [22] propose a signal-based method that exploits the blocking artefacts left by the jpeg compression: Using a pair of high pass filters (once for each image dimension), the traces of DCT quantization are enhanced. Then, using an ad hoc metric, the system computes a measure of the blockiness effect, and finally the starting location of the blocking artifact is found as a 2D vector. If a value greater than zero is found, a crop is detected.

As anticipated earlier, we do not know if the YDB, PKU and TVPD dataset include native or pre-processed images. Therefore, in order to conduct a fair comparison with [22], we built a new dataset, named Florence Building (FLB), composed of 94 images acquired with a Canon 5D Mark II camera. Images are firstly saved in raw cr2 format, then compressed with jpeg using quality factors (QF) 50 and 90. Then images are cropped as described in Sect. 4.1 using all three cropping percentages (20%, 35%, and 50%), and saved in png to avoid a second compression (in order to match the experimental setup of [22]). Note also that we added a random cropping between 1 and 7 pixels to avoid the production images with dimensions multiple of 8 (in this case the detector of [22] is spoiled). In order to produce a score to be evaluated with a ROC curve, given the peak location (p_x, p_y) we compute a score $s = \max(p_x, p_y)$. The *Native* columns of Tab. 10 reports the obtained AUC (ROC plots can be found in the additional material). Note that, for sake of brevity, we included results only for our $\mathcal{D}_{15\%}$.

Table 10: Comparison between the proposed AMARCORD detector and the solution of Bruna et al. [22].

		Native		**Recompressed**	
Metric	**Crop%**	**Qf 50**	**Qf 90**	**Qf 50**	**Qf 90**
AMARCORD ($\mathcal{D}_{15\%}$)	20%	0.5905	0.6441	0.6637	0.6692
	35%	0.7072	0.7478	0.7590	0.7736
	50%	0.7796	0.8112	0.8372	0.8439
Bruna et al.	20%	1.0000	0.3909	0.5150	0.5208
	35%	1.0000	0.4003	0.4800	0.4892
	50%	1.0000	0.4526	0.4700	0.4976

Then, we exchanged all the images throughFacebook, so as to obtain images recompressed by the social network. Results are shown in the *Recompressed* columns of Tab. 10 (for ROC plots see the additional material). As it can be seen, while the method of Bruna et al. [22] is perfect in detecting cropped images with a single compression with low QF (i.e. QF = 50), in case of lighter compression (i.e. QF = 90) or after recompression its performance drops dramatically. On the other hand, AMARCORD produces almost stable results for any testing condition, demonstrating greater effectiveness and robustness on realistic scenarios.

4.10. Evaluation of the opt-out criteria

In this section we present results obtained after including the two heuristic opt-out criteria presented in Sect. 3.4, namely *MaxAngle* and *MaxDist*. Table 11 summarizes the results obtained on all datasets by setting the thresholds to their theoretical values $Th_{MaxAng} = 90$ and $Th_{MaxDist} = 0.5$. AUCs improve both for $\mathcal{D}_2$ and $\mathcal{D}_{15\%}$ for all cropping percentages and for all the Manhattan-world datasets. On the other hand, as a result of opt-out analysis, some of the dataset images were labelled as intractable, with a number of discarded images increasing with the amount of cropping. Notice that opt-out criteria are not only good at improving performance on man-made data (by reducing the false alarm rate), but they are also quite effective in discarding natural images, with correct detection rates of more than 90%. Table 12 reports the results obtained by slightly relaxing the opt-out thresholds from their theoretical values. Note that, since after opt-out very few Natural images remain, we do not compute the related AUCs, completely unreliable to assess performance. The new set of thresholds, namely $Th_{MaxAng} = 95$ and $Th_{MaxDist} = 0.7$, has the beneficial effect of reducing the number of discarded images on man-made datasets, virtually without any loss of AUC performance. The natural image rejection rate is not affected by the change of thresholds.

Table 11: Cropping detection AUCs, obtained with $\mathcal{D}_2$ and $\mathcal{D}_{15\%}$, and percentage of discarded probes using $Th_{MaxAng} = 90$ and $Th_{MaxDist} = 0.5$. Note that the percentage of discarded images is related to both metrics.

Metric	**Crop%**	**YDB**	**PKU**	**TVPD**	**NAT**
$\mathcal{D}_2$	20%	0.6513	0.7079	0.7404	—
	35%	0.8286	0.8244	0.8953	—
	50%	0.9341	0.8470	0.9000	—
$\mathcal{D}_{15\%}$	20%	0.7335	0.7237	0.7623	—
	35%	0.8564	0.8610	0.8977	—
	50%	0.9349	0.8985	0.8893	—
Discarded(%)	20%	39%	47%	30%	98%
	35%	44%	54%	40%	95%
	50%	58%	63%	53%	96%

Table 12: Cropping detection AUCs, obtained with $\mathcal{D}_2$ and $\mathcal{D}_{15\%}$, and percentage of discarded probes using $Th_{MaxAng} = 95$ and $Th_{MaxDist} = 0.7$. Note that the percentage of discarded images is related to both metrics.

Metric	**Crop%**	**YDB**	**PKU**	**TVPD**	**NAT**
$\mathcal{D}_2$	20%	0.6433	0.6963	0.7102	—
	35%	0.8036	0.8106	0.8622	—
	50%	0.9034	0.8390	0.8814	—
$\mathcal{D}_{15\%}$	20%	0.7300	0.7044	0.7425	—
	35%	0.8409	0.8495	0.8810	—
	50%	0.9257	0.8870	0.8838	—
Discarded(%)	20%	23%	36%	23%	97%
	35%	32%	42%	29%	94%
	50%	47%	53%	43%	95%

5. Conclusions and Future Work

This paper presents a fully automated cropping detector for Manhattan scenes, based on the estimation of the camera principal point. Line segments are detected and clustered to locate three dominant vanishing points in the scene using computer vision techniques, and eventually estimate the principal point. A new metric, referred to $\mathcal{D}_{p\%}$, based on a Monte Carlo analysis and taking into account the statistical distribution of the principal point regarded as a random variable, is also introduced and discussed. Moreover, heuristic opt-out criteria for improving the method reliability are proposed and evaluated.

Experimental results on several different datasets show the effectiveness of the proposed framework. In particular, $\mathcal{D}_{15\%}$ achieves the best results, improving significantly over the standard $\mathcal{D}_2$ metric. Also, our solution ehibited a high degree of robustness against counter-forensics attacks (e.g. recompression, enhancement, and filtering), differently from signal-based method for cropping detection that are spoiled by such operations. Additionally, results on opt-out testing demonstrate the effectiveness of the heuristic criteria at improving performance and rejecting intractable images, such as those containing natural scenes.

Future work will be devoted to improving the method performance by increasing the selectivity of the feature extraction (line detection and clustering) stage, since as shown in the tests, using a smarter line selection would be highly beneficial to our approach.

Acknowledgments

This material is based on research partially sponsored by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0188. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing

the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

Appendix: AMARCORD score as a function of the cropping factor

Let I be a pristine image with dimension (W, H). After cropping with a cropping factor $\alpha \in [0, 1[$ we obtain the cropped image I_c with dimensions (w, h) such that

$$W = \frac{w}{1-\alpha} \qquad H = \frac{h}{1-\alpha} \tag{.1}$$

Supposing to work with error free data, the principal point is fixed as $\mathbf{p} = (W/2, H/2)$, while the center of the cropped image is $\mathbf{c} = (w/2, h/2)$. Then, the score computed by AMARCORD, normalized by the cropped image diagonal, is

$$\mathcal{S} = \frac{\mathcal{D}_2(\mathbf{c}, \mathbf{p})}{\sqrt{w^2 + h^2}} = \frac{\sqrt{(w/2 - W/2)^2 + (h/2 - H/2)^2}}{\sqrt{w^2 + h^2}}$$

Using the expressions above for W and H in the score equation, we can express the score as a function of the cropping factor as

$$\begin{aligned}
\mathcal{S}(\alpha) &= \sqrt{\frac{\left(\frac{w - \frac{w}{(1-\alpha)}}{2}\right)^2 + \left(\frac{h - \frac{h}{(1-\alpha)}}{2}\right)^2}{w^2 + h^2}} = \\
&= \sqrt{\frac{\left(\frac{-\alpha w}{2(1-\alpha)}\right)^2 + \left(\frac{-\alpha h}{2(1-\alpha)}\right)^2}{w^2 + h^2}} = \\
&= \sqrt{\frac{\alpha^2 w^2 + \alpha^2 h^2}{4(1-\alpha)^2 (w^2 + h^2)}} = \\
&= \sqrt{\frac{\alpha^2}{4(1-\alpha)^2}}
\end{aligned}$$

from which we finally get

$$\mathcal{S}(\alpha) = \frac{\alpha}{2(1-\alpha)}$$

The cropping score is 0 for pristine images ($\alpha = 0$), and goes to infinity for a cropping factor $\alpha \to 1$.

References

[1] H. Farid, A survey of image forgery detection, IEEE Signal Processing Magazine 26(2) (2009) 16–25.

[2] M. Stamm, M. Wu, K. Liu, Information forensics: An overview of the first decade, IEEE Access 1 (2013) 167–200.

[3] G. K. Birajdar, V. H. Mankar, Digital image forgery detection using passive techniques: A survey, Digital Investigation 10 (3) (2013) 226 – 245.

[4] A. Piva, An overview on image forensics, ISRN Signal Processing 2013 (2013) Article ID 496701, 22 pages.

[5] M. A. Qureshi, M. Deriche, A bibliography of pixel-based blind image forgery detection techniques, Signal Processing: Image Communication 39 (2015) 46 – 74.

[6] V. Schetinger, M. Iuliani, A. Piva, M. M. Oliveira, Image forgery detection confronts image composition, Computers & Graphics 68 (2017) 152–163.

[7] T. Kumar, G. Khurana, Towards recent developments in the field of digital image forgery detection, Int. J. Comput. Appl. Technol. 58 (1) (2018) 1–16.

[8] P. Ferrara, T. Bianchi, A. De Rosa, A. Piva, Image forgery localization via fine-grained analysis of cfa artifacts, IEEE Transactions on Information Forensics and Security 7 (5) (2012) 1566–1577. doi:10.1109/TIFS.2012.2202227.

[9] M. Chen, J. Fridrich, M. Goljan, J. Lukas, Determining image origin and integrity using sensor noise, IEEE Transactions on Information Forensics and Security 3 (1) (2008) 74–90. doi:10.1109/TIFS.2007.916285.

[10] B. Li, T. Ng, X. Li, S. Tan, J. Huang, Revealing the trace of high-quality JPEG compression through quantization noise analysis, IEEE Transactions on Information Forensics and Security 10 (3) (2015) 558–573. doi:10.1109/TIFS.2015.2389148.

[11] T. Bianchi, A. Piva, Image forgery localization via block-grained analysis of jpeg artifacts, IEEE Transactions on Information Forensics and Security 7 (3) (2012) 1003–1017. doi:10.1109/TIFS.2012.2187516.

[12] D. Bhardwaj, V. Pankajakshan, A jpeg blocking artifact detector for image forensics, Signal Processing: Image Communication 68 (2018) 155 – 161.

[13] E. Kee, J. F. O'Brien, H. Farid, Exposing photo manipulation with inconsistent shadows, ACM Trans. Graph. 32 (3) (2013) 28:1–28:12. doi:10.1145/2487228.2487236.

[14] T. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, A. de Rezende Rocha, Exposing digital image forgeries by illumination color classification, IEEE Transactions on Information Forensics and Security (2013) 1182–1194.

[15] M. Johnson, H. Farid, Exposing digital forgeries in complex lighting environments, IEEE Transactions on Information Forensics and Security 2 (3) (2007) 450 –461. doi:10.1109/TIFS.2007.903848.

[16] T. Carvalho, H. Farid, E. Kee, Exposing photo manipulation from user-guided 3d lighting analysis, in: Proc. SPIE, Vol. 9409, 2015, pp. 940902–940902–10.

[17] H. Yao, S. Wang, Y. Zhao, X. Zhang, Detecting image forgery using perspective constraints, Signal Processing Letters, IEEE 19 (3) (2012) 123–126. doi:10.1109/LSP.2011.2182191.

[18] M. Iuliani, G. Fabbri, A. Piva, Image splicing detection based on general perspective constraints, in: Proceedings of the Information Forensics and Security (WIFS), 2015 IEEE International Workshop, 2015.

[19] B. Peng, W. Wang, J. Dong, T. Tan, Position determines perspective: Investigating perspective distortion for image forensics of faces, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1813–1821. `doi:10.1109/CVPRW.2017.227`.

[20] B. Peng, W. Wang, J. Dong, T. Tan, Image forensics based on planar contact constraints of 3d objects, IEEE Transactions on Information Forensics and Security 13 (2) (2018) 377–392. `doi:10.1109/TIFS.2017.2752728`.

[21] M. Zampoglou, S. Papadopoulos, Y. Kompatsiaris, Detecting image splicing in the wild (web), in: Proc. IEEE Int Multimedia & Expo Workshops (ICMEW) Conf, 2015, pp. 1–6.

[22] A. Bruna, G. Messina, S. Battiato, Crop detection through blocking artefacts analysis, in: International Conference on Image Analysis and Processing, Springer, 2011, pp. 650–659.

[23] W. Li, Y. Yuan, N. Yu, Passive detection of doctored jpeg image via block artifact grid extraction, Signal Processing 89 (9) (2009) 1821–1829.

[24] X. Meng, S. Niu, R. Yan, Y. Li, Detecting photographic cropping based on vanishing points, Chinese Journal of Electronics 22 (2013) Article ID 496701, 22 pages.

[25] M. Iuliani, M. Fanfani, C. Colombo, A. Piva, Reliability assessment of principal point estimates for forensic applications, Journal of Visual Communication and Image Representation 42 (2017) 65–77.

[26] J. M. Coughlan, A. L. Yuille, Manhattan world: compass direction from a single image by bayesian inference, in: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, Vol. 2, 1999, pp. 941–947 vol.2.

[27] R. I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, 2004.

[28] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334.

[29] R. Szeliski, Computer Vision: Algorithms and Applications, 1st Edition, Springer-Verlag New York, Inc., New York, NY, USA, 2010.

[30] C. Colombo, D. Comanducci, A. Del Bimbo, Camera Calibration with Two Arbitrary Coaxial Circles, in: Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 265–276.

[31] E. Guillou, D. Meneveaux, E. Maisel, K. Bouatouch, Using vanishing points for camera calibration and coarse 3d reconstruction from a single image, The Visual Computer 16 (7) (2000) 396–410.

[32] J. Deutscher, M. Isard, J. Maccormick, Automatic camera calibration from a single manhattan image, in: Eur. Conf. on Computer Vision (ECCV, 2002, pp. 175–205.

[33] R. Pflugfelder, H. Bischof, Online auto-calibration in man-made worlds, in: Digital Image Computing: Techniques and Applications (DICTA'05), 2005, pp. 75–75.

[34] M. K. Johnson, H. Farid, Detecting photographic composites of people., in: Y. Q. Shi, H.-J. Kim, S. K. 0001 (Eds.), IWDW, Vol. 5041 of Lecture Notes in Computer Science, Springer, 2007, pp. 19–33.

[35] Y. Li, Y. Zhou, K. Yuan, Y. Guo, X. Niu, Exposing photo manipulation with inconsistent perspective geometry, The Journal of China Universities of Posts and Telecommunications 21 (4) (2014) 83 – 104.

[36] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (6) (1986) 679–698.

[37] J. P. Tardif, Non-iterative approach for fast and accurate vanishing point detection, in: 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 1250–1257.

[38] R. Toldo, A. Fusiello, Robust multiple structures estimation with j-linkage, in: Proceedings of the 10th European Conference on Computer Vision: Part I, ECCV '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 537–547.

[39] D. Row, T. Reid, Geometry, Perspective Drawing, and Mechanisms, World Scientific, 2012.

[40] P. Denis, J. H. Elder, F. J. Estrada, Efficient edge-based methods for estimating manhattan frames in urban imagery, in: D. Forsyth, P. Torr, A. Zisserman (Eds.), Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II, Springer Berlin Heidelberg, 2008, pp. 197–210.
URL `http://www.elderlab.yorku.ca/YorkUrbanDB/`

[41] B. Li, K. Peng, X. Ying, H. Zha, Simultaneous vanishing point detection and camera calibration from single images, in: Advances in Visual Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 151–160.
URL `http://www.cis.pku.edu.cn/vision/vpdetection/`

[42] V. Angladon, S. Gasparini, V. Charvillat, The toulouse vanishing points dataset, in: Proceedings of the 6th ACM Multimedia Systems Conference (MMSys '15), Portland, OR, United States, 2015. `doi:10.1145/2713168.2713196`.
URL `http://ubee.enseeiht.fr/dokuwiki/doku.php?id=public:toulousevpdataset`

[43] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, A. Piva, Vision: a video and image dataset for source identification, EURASIP Journal on Informa-

tion Security 2017 (1) (2017) 15. doi:10.1186/s13635-017-0067-2.
URL ftp://lesc.dinfo.unifi.it/pub/Public/VISION/