Manifold Learning for Super Resolution

Von der Fakultät für Elektrotechnik und Informatik

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte

Dissertation

von

M. Sc. Eduardo Pérez Pellitero

geboren am 5. November 1989 in Barcelona.

Betreuer:
Prof. Bodo Rosenhahn, Leibniz Universität Hannover

Gutachter:
Prof. Jörn Ostermann, Leibniz Universität Hannover
Prof. Javier Ruiz-Hidalgo, Universitat Politècnica de Catalunya

Verteidigung der Dissertation: **21.02.2017**

Acknowledgement

Part of the joy of research has a lot to do with the people with whom you share it. I would like to specially thank my doctoral advisor Prof. Dr.-Ing Bodo Rosenhahn for his unconditional support and the always interesting research discussions. Prof. Javier Ruiz has also helped me a lot by always posing challenging questions, being critical but at the same time supportive. I would also like to thank Prof. Dr.-Ing Jörn Ostermann for being part of the defense tribunal.

I keep very good memories from my time at Technicolor, and inevitably some names are linked to them. Special thanks to my research supervisor Dr. Jordi Salvador, who guide me closely during my first steps in research and pushed me always towards the *sunny side* of science. Thanks also to Axel Kochale for being such a helpful person and for the countless times I heard his laughter across the corridor and made me feel a bit happier.

Last but not least, I feel deeply thankful to my parents, who were always there for me, and to my brother Aitor for being such a great brother to look up to.

Contents

Abbreviations

ANR	Anchored Neighborhood Regression
ASRF	Super-Resolution Forest with alternative training
CS	Cosine Similarity
DCT	Discrete Cosine Transform
DLT	Dense Local Training
GIBP	Gradient Iterative Back Projection
GR	Global Regressor
GSM	Gaussian Scale Mixture
HHC	Half-Hypersphere Confinement
HR	High-Resolution
IBP	Iterative Back Projection
IFC	Information Fidelity Criterion
LASSO	Least Absolute Shrinkage and Selection Operator
LR	Low-Resolution
MAP	Maximum-a-posteriori
MDS	Multidimensional Scaling
MF	Magnification Factor
ML	Maximum Likelihood
MSE	Mean Squared Error
NBNN	Naive Bayes Nearest Neighbor
NBSRF	Naive Bayes Super-Resolution Forest
NN	Nearest Neighbor
OMP	Orthogonal Matching Pursuit
PCA	Principal Component Analysis
PSNR	Peak Signal-to-Noise Ratio
PSyCo	Patch Symmetry Collapse
SCT	Symmetry-Collapsing Transform

SD	Symmetry Distance
SIFT	Scale Invariant Feature Transform
SpH	Spherical Hashing
SR	Super Resolution
SRCNN	Super Resolution using Convolutional Neuronal Networks
SSIM	Structural Similarity
ST	Symmetry Transform
SVD	Singular Value Decomposition
VQ	Vector Quantization

Abstract

The development pace of high-resolution displays has been so fast in the recent years that many images acquired with low-end capture devices are already outdated or will be shortly in time. Super Resolution is central to match the resolution of the already existing image content to that of current and future high resolution displays and applications. This dissertation is focused on *learning* how to upscale images from the statistics of natural images. We build on a sparsity model that uses learned coupled low- and high-resolution dictionaries in order to upscale images.

Firstly, we study how to adaptively build coupled dictionaries so that their content is semantically related with the input image. We do so by using a Bayesian selection stage which finds the best-fitted texture regions from the training dataset for each input image. The resulting adapted subset of patches is compressed into a coupled dictionary via sparse coding techniques.

We then shift from ℓ_1 to a more efficient ℓ_2 regularization, as introduced by Timofte et al. [74]. Instead of using their patch-to-dictionary decomposition, we propose a fully collaborative neighbor embedding approach. In this novel scheme, for each atom in the dictionary we create a densely populated neighborhood from an extensive training set of raw patches (i.e. in the order of hundreds of thousands). This generates more accurate regression functions.

We additionally propose using sublinear search structures such as spherical hashing and trees to speed up the nearest neighbor search involved in regression-based Super Resolution. We study the positive impact of antipodally invariant metrics for linear regression frameworks, and we propose two efficient solutions: (a) the Half Hypersphere Confinement, which enables antipodal invariance within the Euclidean space, and (b) the bimodal tree, whose split functions are designed to be antipodally invariant and which we use in the context of a Bayesian Super Resolution forest.

In our last contribution, we extend antipodal invariance by also taking into consideration the dihedral group of transforms (i.e. rotations and reflections). We study them as a group of symmetries within the high-dimensional manifold. We obtain the respective set of mirror-symmetry axes by means of a frequency analysis, and we use them to collapse the redundant variability, resulting in a reduced manifold span which, in turn, greatly improves quality performance and reduces the dictionary sizes.

Kurzfassung

In den letzten Jahren ist die Entwicklung von hochauflösenden Displays so rasant verlaufen, dass viele niedrigauflösende Bildaufnahmegeräte entweder schon überholt sind oder es bald sein werden. Super Resolution ist essentiell, um die Auflösung der existierenden Bilder, bzw. Bildinhalte auf die aktuellen und zukünftigen hochauflösenden Displays und Apps zu übertragen. Diese Dissertation beschäftigt sich mit dem lernen des Upscalings von Bildern aus Statistiken realer Bilder. Die Basis hierbei bildet ein *sparsity-Modell*, dass bereits gelernte, gekoppelte hoch- und niedrig auflösende Wörterbücher verwendet, um damit Bilder hochzuskalieren.

Als erstes beschäftigt sich diese Arbeit damit, wie man gekoppelte Wörterbücher adaptativ aufbaut, so, dass ihr Inhalt semantisch mit dem jeweiligen Eingangsbild verknüpft wird. Dies geschieht, indem eine Bayes-Auswahlstufe verwendet wird, welche die am besten passenden Texturbereiche aus dem Trainings-Datensatz für jedes neue Eingangsbild auswählt. Das daraus resultierende angepasste Untermenge von Patches wird anschließend über das Sparse-Coding Techniken in ein gekoppeltes Wörterbuch komprimiert.

Statt einer ℓ_1 Regularisierung, nutzen wir die effizientere ℓ_2 Regularisierung, wie von Timofte et al. [74] vorgeschlagen. Anstatt ihrer *patch-to-dictionary*-Zerlegung, wird ein vollstandig kollaborativer Ansatz zur Nachbarschaftseinbettung vorgeschlagen. In diesem neuen Modell schaffen wir für jedes Atom des Wörterbuchs eine dicht besetzte Nachbarschaft die sich aus dem vorherigen umfangreichen Trainingsdatensatz der Roh-Patches ergibt. Dies generiert genauere Regressionsfunktionen.

Zusätzlich wird vorgeschlagen eine sublineare Suchstruktur, z. B. *sperical hashing* und *Bäume* zu nutzen, um die Suche nach dem nächsten Nachbarn zu beschleunigen, die in der regressions-basierten Super-Resolution genutzt wird. Wir analysieren die positiven Auswirkungen der antipodal invarianten Metriken für lineare Regression-frameworks und zwei effiziente Lösungen werden vorgeschlagen: a) das *Half-Hypersphere Confinement*, was die antipode Invarianz innerhalb des Euklidischen Raums ermöglicht sowie b) den *bimodal Baum*, dessen gesplittete Funktionen so angelegt sind, dass sie antipodal invariant sind. Diese werden im dann im Kontext eines „Bayeschen Super Resolution forest“ angewendet.

Als letzten Beitrag wird die antipodale Invarianz ausgeweitet, indem die Diedergruppe der Transformationen (d.h. Rotationen und Reflexionen) ebenso mit einbezogen werden. Diese werden als eine Symmetriegruppe in der hochdimensionalen Mannigfaltigkeit der Patches genauer betrachtet. Die jeweiligen Spiegelsymmetrieachsen werden durch eine Frequenzanalyse bestimmt und anschließend verwendet, um die redundanten Variabilitäten zusammenzufassen. Dies mündet in einer Reduktion des Spannraums der Mannigfaltigkeit, was in einer verbesserten Performanzqualität resultiert und gleichzeitig die Größe des Wörterbuchs verkleinert.

Chapter 1

Introduction

1.1 Problem statement

Applications delivering low resolution images are diverse (e.g. surveillance, satellite, live streaming) and there is also abundant multimedia content whose resolution is not up-to-date with current display capabilities. To fill in this gap, Super Resolution (SR) techniques are used. SR aims to obtain a high-resolution image from its degraded low-resolution observation, eventually surpassing the limits of the original capture device. SR deals with a deeply ill-posed problem, and therefore requires further constraints or prior knowledge in order to make the problem tractable.

SR is usually differentiated from other simpler upscaling methods (e.g. bilinear, bicubic [38], Lanczos [80] interpolation) by its high-quality performance, which yields not only lower objective error measurements, but also more natural and pleasant images. Therefore, we can arguably state that quality is the leading characteristic of SR. Nonetheless, there are also other factors of major importance for the successful wide adoption of these techniques. SR as an end application requires processing considerable amounts of data due to the ongoing shift to higher frame rates and spatial resolutions in video content. When used as a pre-processing step in other computer vision problems (e.g. object detection, object classification [12, 50]), the access to computing resources is limited and the processing times should not cause great impact in the overall pipeline. For both situations, speed and memory size becomes crucial.

In this thesis we tackle the single-image SR problem aiming to improve the aforementioned three-fold defining factors: quality, speed and memory usage.

1.2 Motivation

The application of SR to images has been a very active research field in the recent years, with multiple applications involving diverse fields. As of today, it has been used in cinema production business when there is a need to adapt content to different cinema specifications (i.e. different resolutions and frame rates [69, 72]), in video-games live streaming [71], in microscopy imaging [34], in photography post-

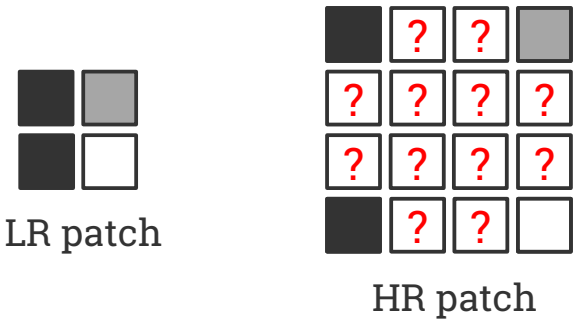


Figure 1.1: SR is an ill-posed problem, i.e. the High-Resolution (HR) patch has fewer constraints than unknown variables and therefore has not a unique solution.

processing [18, 23, 27, 35, 53, 55] and in Magnetic Resonance Imaging [81], just to name a few. We show some examples of the improved quality obtained with SR in Figure 1.2.

Not only SR is useful as an end application, but also it can help other computer vision problems whenever the resolution is not sufficient for e.g. object detection, scene understanding or feature extraction [50, 12]. Even content which is broadcast today by major television channels requires further upscaling, as they can not match the latest displays capabilities (e.g. 4k displays). This tendency is as of today a reality and it is not uncommon to find displays which are already equipped with SR algorithms.

1.3 Challenges

Super Resolution presents many challenges due to its ill-posed nature, i.e. there is not a unique solution for the unknown pixel values of the HR image (see Figure 1.1). The ground of such behavior can be explained by means of the Nyquist-Shannon sampling theorem [46], in which the sampling rate f_s for a perfect reconstruction should be at least two times the maximum frequency of the signal, i.e. $f_s > 2f_{max}$. If this sampling frequency is not fulfilled and the non-compliant frequencies are not filtered out, the resulting image might contain aliasing. Downscaling an image can be seen as decreasing the sampling frequency $f'_s < f_s$. Even when the resulting image does not contain aliased frequencies, if there has been a downscaling operation, a new Nyquist frequency f'_{max} has been imposed and as a result some frequencies from the upper bands have been lost.

When we upscale a previously downscaled degraded image Y, we are not able to recover the data loss that occurred during the downscaling, as those frequencies are beyond the f'_{max}. The common approach in classic image processing is to assume a smooth prior in the reconstruction. This usually translates into averaging over neighboring samples (e.g. bilinear, bicubic functionals), which as expected, produces

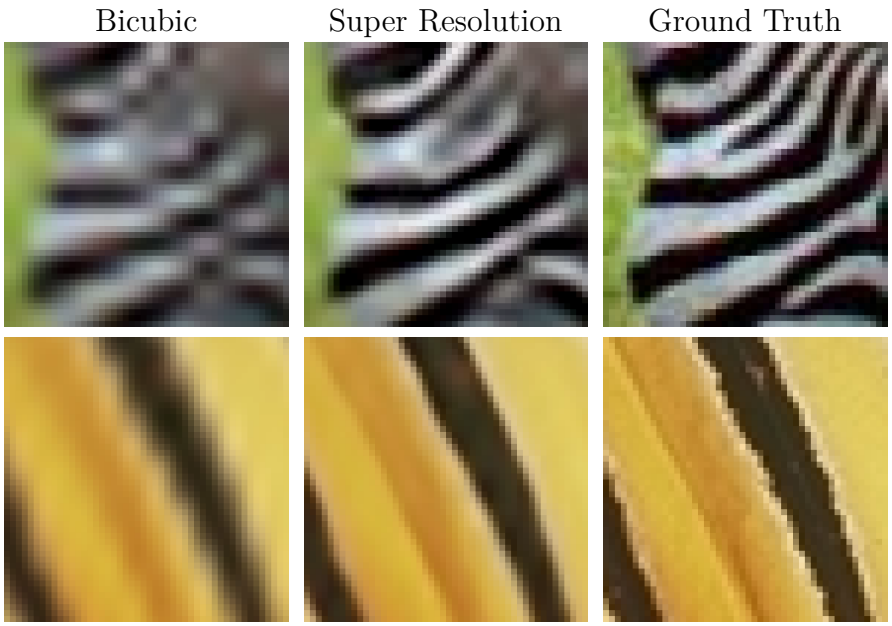


Figure 1.2: Image upscaling using bicubic interpolation and one of our proposed SR algorithms (see details in Chapter 8).

overly smooth solutions, which highly differ from the original image X. We show an illustrative example of such behavior in Figure 1.3.

SR aims to improve the prediction of that unknown data by the progressive sophistication of priors. By *learning* the correspondence between degraded Low-Resolution (LR) and HR patches we are able to infer the missing frequencies from the available cues. As other machine learning approaches, example-based SR shares some of the fundamental challenges of statistical learning: designing a learning scheme that does not under- or over-fit and that generalizes properly for all possible patch variations.

1.4 Related work

Early approaches to SR showed that it was possible to reconstruct higher-resolution images by registering and fusing multiple images [79], thus pioneering a vast amount of approaches on multi-image SR, often called reconstruction-based SR. This idea was further refined, among others, with the introduction of iterative back-projection for improved registration by Irani and Peleg [37], although further analysis by Baker and Kanade [2, 3] and Lin and Shum [42, 43, 84] showed fundamental limits on this type of SR, mainly conditioned by registration accuracy. Learning-based SR, also known as example-based, overcame some of the aforementioned limitations by avoiding the necessity of a registration process and by building the priors from image statistics. The original work by Freeman et al. [25, 24] aims to *learn* from patch- or

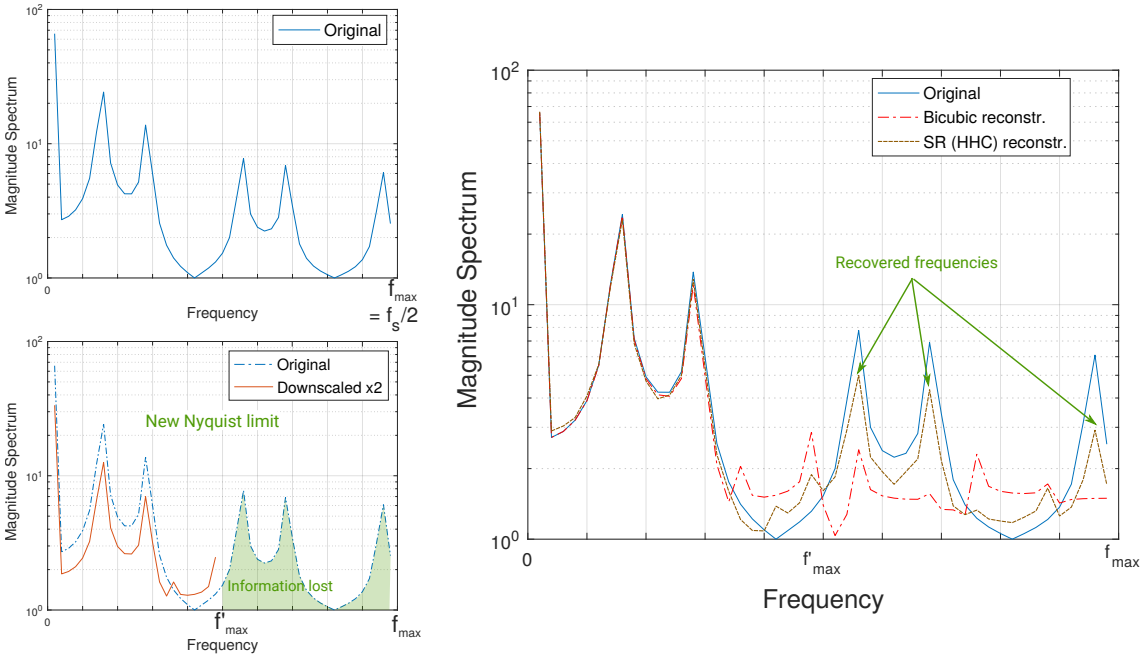


Figure 1.3: Spectrum magnitude for several images containing only horizontal frequencies from an the original image X (top left), the downscaled image Y (bottom left) and the reconstructed HR images via bicubic and SR (right). Due to the new Nyquist frequency imposed during downsampling (f'_{max}) there is several information loss. Bicubic interpolation can not properly reconstruct all that information and generates distorsion in the upper band of frequencies. SR, however, manages to correctly recover several harmonics with lesser distortions.

feature-based examples to produce effective magnification well beyond the practical limits of multi-image SR.

Example-based SR approaches using dictionaries are usually divided into two categories: internal and external dictionary-based SR. The first exploits the strong self-similarity prior. This prior is learnt directly from the relationship of image patches across different scales of the input image [28].

External dictionary-based SR uses other images to build their dictionaries. A representative widely used approach is the one based on sparse decomposition. The main idea behind this approach is the decomposition of each patch in the input image into a combination of a sparse subset of entries in a compact dictionary. The work of Yang et al. [95] uses an external database composed of related low and high-resolution patches to jointly learn a compact dictionary pair. During testing, each image patch is decomposed into a sparse linear combination of the entries in the LR dictionary and the same weights are used to generate the HR patch as a linear combination of the HR entries. The work presented in this thesis builds on the sparsity model of Yang et al. [95] and regression-based approaches described in

more detail in Chapters 2 and 3.

1.5 Contributions

The contributions of this thesis are:

1. We propose an algorithm to build semantically adaptive dictionaries based on the Naive Bayes assumption (Chapter 4).

2. We introduce a dense, local, collaborative ℓ_2-regularized training scheme for regression-based SR which results in improved quality (Chapter 5).

3. We adapt the use of Spherical Hashing to the nearest neighbor search problem of piece-wise linear regression for SR (Chapter 5).

4. We study the positive impact of antipodally invariant metrics for linear regression models and recommend its usage. We propose the efficient Half Hypersphere Confinement transform which embeds antipodal invariant metrics in the Euclidean space and thus enables other search algorithms, e.g. Spherical Hashing, to be antipodally invariant (Chapter 6).

5. We propose a bimodal tree split function which can be used both for unsupervised clustering and fast inference in regression-based SR. We propose to use those bimodal trees within a regression forest, which adaptively selects the best tree for each patch through a Naive Bayes efficient selection stage (Chapter 7).

6. We analyze the patch-manifold symmetries induced by the dihedral group of transforms (i.e. reflections and rotations) and design a low-complexity transform that collapses the symmetries induced by both the dihedral group and the antipodes (Chapter 8).

1.6 Overview

This thesis is organized following an incremental line of contributions. The fundamentals are described in Chapters 2 and 3. The contributions of the thesis are presented in Chapters 4-8.

Each chapter has an introduction and a closing summary and discussion. All the experimental results are included in a single chapter at the end of the dissertation. Some minor validation tests are included in each chapter when necessary for the context of the method description.

The thesis outline is:

- **Chapter 1: Introduction.** Problem statement, motivation, challenges, related work and contributions.
- **Chapter 2: Sparse dictionaries for SR**. We describe the sparse SR model of Yang et al. [94, 95] and how Zeyde et al. [97] improved some of its complexity bottlenecks. This thesis builds on top of the dictionary model presented in this chapter.
- **Chapter 3: Anchored Neighborhood Regression.** We discuss the work of Timofte et al. [74] which is highly relevant to the rest of contributions of this thesis.
- **Chapter 4: Bayesian approach to adaptive dictionaries.** We introduce our first contribution. We propose a Bayesian model to adaptively train sparse dictionaries [60]. This idea is later adapted to be used within a Forest (Chapter 7).
- **Chapter 5: Dense Local Training and Spherical Hashing.** We present a novel training approach based on dense local neighborhoods and a sublinear spherical hashing nearest neighbor search [62].
- **Chapter 6: Half Hypersphere Confinement.** We introduce the importance of antipodally invariant metrics, and propose how to embed them in the Euclidean space through a half hypersphere confinement transform [54, 53].
- **Chapter 7: Naive Bayes SR Forest**. We present a novel SR algorithm based on regression forest whose split functions are antipodally invariant (i.e. bimodal tree). We adapt the Naive Bayes selection stage of Chapter 4 to be able to select a tree within the forest [65].
- **Chapter 8: Dihedral Symmetry Collapse.** We present our latest work on dihedral symmetries of patch manifolds. We propose the Patch Symmetry Collapse in order to reduce the span of the patch manifold in order to have a better-posed learning problem [55].
- **Chapter 9: Results**. We explain our experimental methology, the evaluation metrics used and the testing datasets. Each method is then assessed separately: we discuss its parameters and we compare them to their other relevant methods. In our last section we do a more comprehensive state-of-the-art comparison.
- **Chapter 10: Conclusions.** We close this thesis with a summary of the contributions and some future work lines.

We show a graphical overview of the contents of this thesis in Figure 1.4.

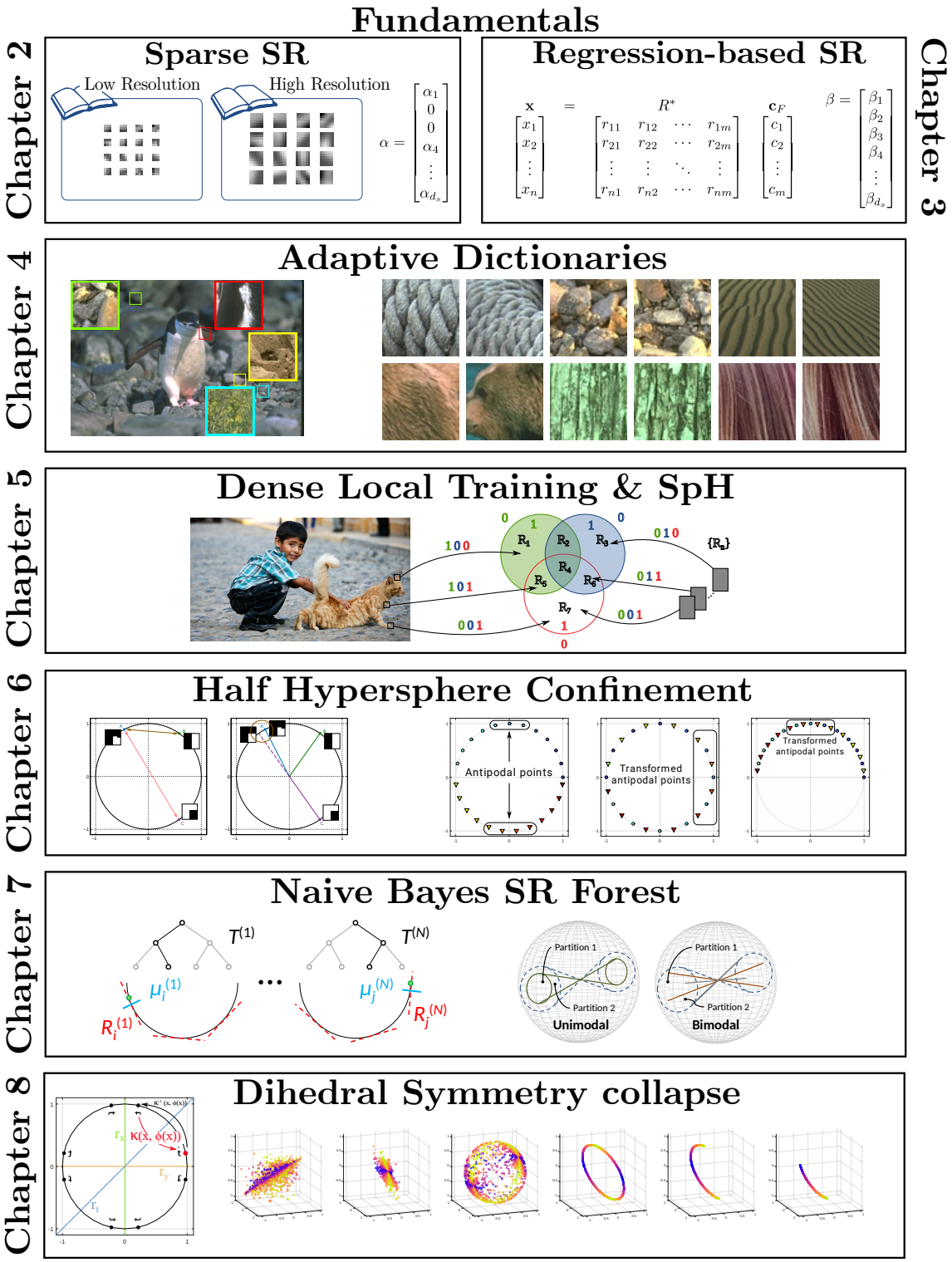


Figure 1.4: Outline of this dissertation.

1.7 Author's papers

The following papers have been published during the lapse of this dissertation and are relevant to it.

[55] **E. Pérez-Pellitero**, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. PSyCo: Manifold span reduction for super resolution. In **CVPR**, 2016.
The main challenge in Super Resolution (SR) is to discover the mapping between the low- and high-resolution manifolds of image patches, a complex ill-posed problem which has recently been addressed through piecewise linear regression with promising results. In this paper we present a novel regression-based SR algorithm that benefits from an extended knowledge of the structure of both manifolds. We propose a transform that collapses the 16 variations induced from the dihedral group of transforms (i.e. rotations, vertical and horizontal reflections) and antipodality (i.e. diametrically opposed points in the unitary sphere) into a single primitive. The key idea of our transform is to study the different dihedral elements as a group of symmetries within the high-dimensional manifold. We obtain the respective set of mirror-symmetry axes by means of a frequency analysis of the dihedral elements, and we use them to collapse the redundant variability through a modified symmetry distance. The experimental validation of our algorithm shows the effectiveness of our approach, which obtains competitive quality with a dictionary of as little as 32 atoms (reducing other methods' dictionaries by at least a factor of 32) and further pushing the state-of-the-art with a 1024 atoms dictionary.

[53] **E. Pérez-Pellitero**, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Antipodally invariant metrics for fast regression-based super-resolution. IEEE **Trans. Image Processing**, 25(6):2456–2468, 2016.
Dictionary-based Super-Resolution algorithms usually select dictionary atoms based on distance or similarity metrics. Although the optimal selection of nearest neighbors is of central importance for such methods, the impact of using proper metrics for Super-Resolution (SR) has been overlooked in literature, mainly due to the vast usage of Euclidean distance. In this paper we present a very fast regression-based algorithm which builds on densely populated anchored neighborhoods and sublinear search structures. We perform a study of the nature of the features commonly used for SR, observing that those features usually lie in the unitary hypersphere, where every point has a diametrically opposite one, i.e. its antipode, with same module and angle, but opposite direction. Even though we validate the benefits of using antipodally invariant metrics, most of the binary splits use Euclidean distance, which does not handle

antipodes optimally. In order to benefit from both worlds, we propose a simple yet effective Antipodally Invariant Transform (AIT) that can be easily included in the Euclidean distance calculation. We modify the original Spherical Hashing algorithm with this metric in our Antipodally Invariant Spherical Hashing scheme, obtaining the same performance as a pure antipodally invariant metric. We round up our contributions with a novel feature transform that obtains a better coarse approximation of the input image thanks to Iterative Back Projection. The performance of our method, which we named Antipodally Invariant Super-Resolution (AIS), improves quality (PSNR) and it is faster than any other state-of-the-art method.

[54] **E. Pérez-Pellitero**, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Half hypersphere confinement for piecewise linear regression. In **WACV**, 2016.

Recent research in piecewise linear regression for Super-Resolution has shown the positive impact of training regressors with densely populated clusters whose datapoints are tight in the Euclidean space. In this paper we further research how to improve the locality condition during the training of regressors and how to better select them during testing time. We study the characteristics of the metrics best suited for the piecewise regression algorithms, in which comparisons are usually made between normalized vectors that lie on the unitary hypersphere. Even though Euclidean distance has been widely used for this purpose, it is suboptimal since it does not handle antipodal points (i.e. diametrically opposite points) properly, as vectors with same module and angle but opposite directions are, for linear regression purposes, identical. Therefore, we propose the usage of antipodally invariant metrics and introduce the Half Hypersphere Confinement (HHC), a fast alternative to Multidimensional Scaling (MDS) that allows to map antipodally invariant distances in the Euclidean space with very little approximation error. By doing so, we enable the usage of fast search structures based on Euclidean distances without undermining their speed gains with complex distance transformations. The performance of our method, which we named HHC Regression (HHCR), applied to Super-Resolution (SR) improves both in quality (PSNR) and it is faster than any other state-of-the-art method. Additionally, under an application-agnostic interpretation of our regression framework, we also test our algorithm for denoising and depth upscaling with promising results.

[65] J. Salvador and **E. Pérez-Pellitero**. Naive Bayes Super-Resolution Forest. In **ICCV**, 2015.

This paper presents a fast, high-performance method for super resolution with external learning. The first contribution leading to the excellent performance is a bimodal tree for clustering, which successfully exploits the antipodal invariance of the coarse-to-high-res mapping of natural image patches and provides scalability to finer partitions of the underlying coarse patch space. During training an ensemble of such bimodal trees is computed, providing different linearizations of the mapping. The second and main contribution is a fast inference algorithm, which selects the most suitable mapping function within the tree ensemble for each patch by adopting a Local Naive Bayes formulation. The resulting method is beyond one order of magnitude faster and performs objectively and subjectively better than the current state of the art.

[62] **E. Pérez-Pellitero**, J. Salvador, I. Torres, Javier Ruiz-Hidalgo, and Bodo Rosenhahn. Fast super-resolution via dense local training and inverse regressor search. In **ACCV**, 2014.
Regression-based Super-Resolution (SR) addresses the upscaling problem by learning a mapping function (i.e. regressor) from the low-resolution to the high-resolution manifold. Under the locally linear assumption, this complex non-linear mapping can be properly modeled by a set of linear regressors distributed across the manifold. In such methods, most of the testing time is spent searching for the right regressor within this trained set. In this paper we propose a novel inverse-search approach for regression-based SR. Instead of performing a search from the image to the dictionary of regressors, the search is done inversely from the regressors' dictionary to the image patches. We approximate this framework by applying spherical hashing to both image and regressors, which reduces the inverse search into computing a trained function. Additionally, we propose an improved training scheme for SR linear regressors which improves perceived and objective quality. By merging both contributions we improve speed and quality compared to the state-of-the-art.

[60] **E. Pérez-Pellitero**, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Bayesian region selection for adaptive dictionary-based super-resolution. In **BMVC**, 2013.
The performance of dictionary-based super-resolution (SR) strongly depends on the contents of the training dataset. Nevertheless, many dictionary-based SR methods randomly select patches from of a larger set of training images to build their dictionaries, thus relying on patches being diverse enough. This paper describes a dictionary building method for SR based on adaptively selecting an optimal subset of patches out of the training images. Each training image is divided into sub-image entities,

named regions, of such a size that texture consistency is preserved and high-frequency (HF) energy is present. For each input patch to super-resolve, the best-fitting region is found through a Bayesian selection. In order to handle the high number of regions in the training dataset, a local Naive Bayes Nearest Neighbor (NBNN) approach is used. Trained with this adapted subset of patches, sparse coding SR is applied to recover the high-resolution image. Experimental results demonstrate that using our adaptive algorithm produces an improvement in SR performance with respect to non-adaptive training.

Other papers have been published during the time frame of the thesis with lesser relevance to this dissertation:

[61] **E. Pérez-Pellitero**, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Accelerating super-resolution for 4k upscaling. In **ICCE**, 2015.
This paper presents a fast Super-Resolution (SR) algorithm based on a selective patch processing. Motivated by the observation that some regions of images are smooth and unfocused and can be properly upscaled with fast interpolation methods, we locally estimate the probability of performing a degradation-free upscaling. Our proposed framework explores the usage of supervised machine learning techniques and tackles the problem using binary boosted tree classifiers. The applied upscaler is chosen based on the obtained probabilities: (1) A fast upscaler (e.g. bicubic interpolation) for those regions which are smooth or (2) a linear regression SR algorithm for those which are ill-posed. The proposed strategy accelerates SR by only processing the regions which benefit from it, thus not compromising quality. Furthermore all the algorithms composing the pipeline are naturally parallelizable and further speed-ups could be obtained.

[77] I. Torres, J. Salvador, and **E. Pérez-Pellitero**. Fast approximate nearest-neighbor field by cascaded spherical hashing. In **ACCV**, 2014.
We present an efficient and fast algorithm for computing approximate nearest neighbor fields between two images. Our method builds on the concept of Coherency-Sensitive Hashing (CSH), but uses a recent hashing scheme, Spherical Hashing (SpH), which is known to be better adapted to the nearest-neighbor problem for natural images. Cascaded Spherical Hashing concatenates different configurations of SpH to build larger Hash Tables with less elements in each bin to achieve higher selectivity. Our method is able to amply outperform existing techniques like PatchMatch and CSH. The parallelizable scheme has been straightforwardly implemented in OpenCL, and the experimental results show that our algorithm is faster and more accurate than existing methods.

[66] J. Salvador, **E. Pérez-Pellitero**, and A. Kochale. Robust Single-Image Super-Resolution using Cross-Scale Self-Similarity. In **ICIP**, 2014.
We present a noise-aware single-image super-resolution (SI-SR) algorithm, which automatically cancels additive noise while adding detail learned from lower-resolution scales. In contrast with most SI-SR techniques, we do not assume the input image to be a clean source of examples. Instead, we adapt the recent and efficient in-place cross-scale self-similarity prior for both learning fine detail examples and reducing image noise. Our experiments show a promising performance, despite the relatively simple algorithm. Both objective evaluations and subjective validations show clear quality improvements when upscaling noisy images.

[9] I. Bosch, J. Salvador, **E. Pérez-Pellitero**, and J. Ruiz-Hidalgo. An epipolar-constrained prior for efficient search in multi-view scenarios. In **EUSIPCO**, 2014.
In this paper we propose a novel framework for fast exploitation of multi-view cues with applicability in different image processing problems. In order to bring our proposed framework into practice, an epipolar-constrained prior is presented, onto which a random search algorithm is proposed to find good matches among the different views of the same scene. This algorithm includes a generalization of the local coherency in 2D images for multi-view wide-baseline cases. Experimental results show that the geometrical constraint allows a faster initial convergence when finding good matches. We present some applications of the proposed framework on classical image processing problems.

[67] J. Salvador, **E. Pérez-Pellitero**, and A. Kochale. Fast single-image super-resolution with filter selection. In **ICIP**, 2013.
This paper presents a new method for estimating a super-resolved version of an observed image by exploiting cross-scale self-similarity. We extend prior work on single-image super-resolution by introducing an adaptive selection of the best fitting upscaling and analysis filters for example learning. This selection is based on local error measurements obtained by using each filter with every image patch, and contrasts with the common approach of a constant metric in both dictionary-based and internal learning super-resolution. The proposed method is suitable for interactive applications, offering low computational load and a parallelizable design that allows straight-forward GPU implementations. Experimental results also show how our method generalizes better to different datasets than dictionary-based super-resolution and comparably to internal learning with adaptive post-processing.

Chapter 2

Sparse dictionaries for SR

2.1 Introduction

Within the familiy of example-based SR algorithms, the notion of using a set of two related dictionaries (LR and HR dictionaries) has been extensively adopted in order to capture the relationship between the LR and HR patches. In recent years, one of the algorithm that popularized dictionaries the most has been the one based on the sparsity prior, first introduced by Yang et al. [94, 95].

These algorithms are based on the sparse signal representation research, i.e. patches can be represented as a sparse linear combination of properly selected atoms from an overcomplete dictionary [41]. The impact of both algorithms in the research community has been notable, as they triggered several strict follow-ups [60, 97, 99, 45], but also because they laid the ground for other dictionary- and regression-based methods that depart from the sparsity prior while still keeping certain common elements [62, 53, 54, 55, 74, 76]. In this section we review the original sparse SR algorithm of Yang et al. [95] and the Zeyde et al. [97] follow-up that addressed some of the limiting factors of its predecessor.

2.2 Model for Sparse SR

Given a LR image Y the single-image SR algorithm of Yang et al. [94, 95] aims to recover a higher-resolution image X by means of two constrains: the reconstruction constraint (also named generation model constraint) and the sparsity prior. In the first one, the observed LR image Y is obtained through a downsampling and blurring of the original image X:

$$Y = S \downarrow (H * X), \tag{2.1}$$

where $S \downarrow$ denotes a downsampling operator of factor S and H is a blurring filter (which could be estimated, but is normally assumed to be bicubic filtering [94, 95, 92, 93, 97, 74, 76, 75]). We show an schematic of the reconstruction constrain and how SR fits on it in Figure 2.1.

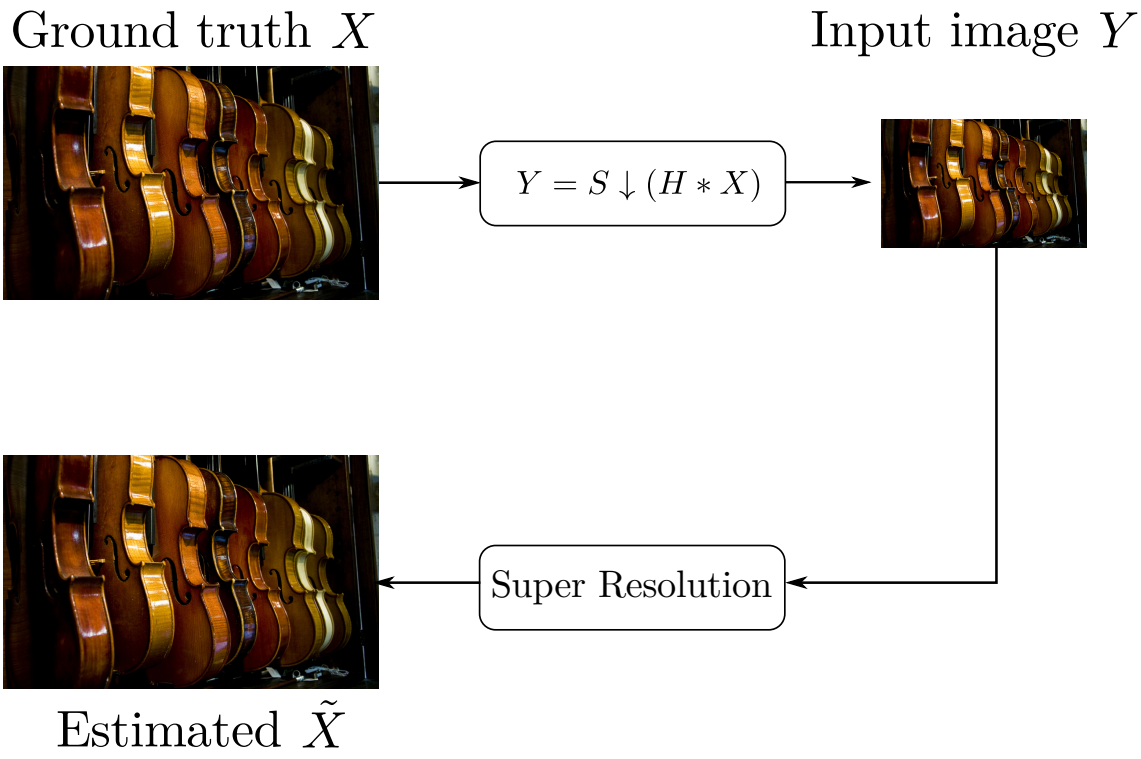


Figure 2.1: Reconstruction constrain of Equation (2.1). The original image X is downscaled and blurred (Y image). Super Resolution aims at reverting this degradation and estimates $\tilde{X}$.

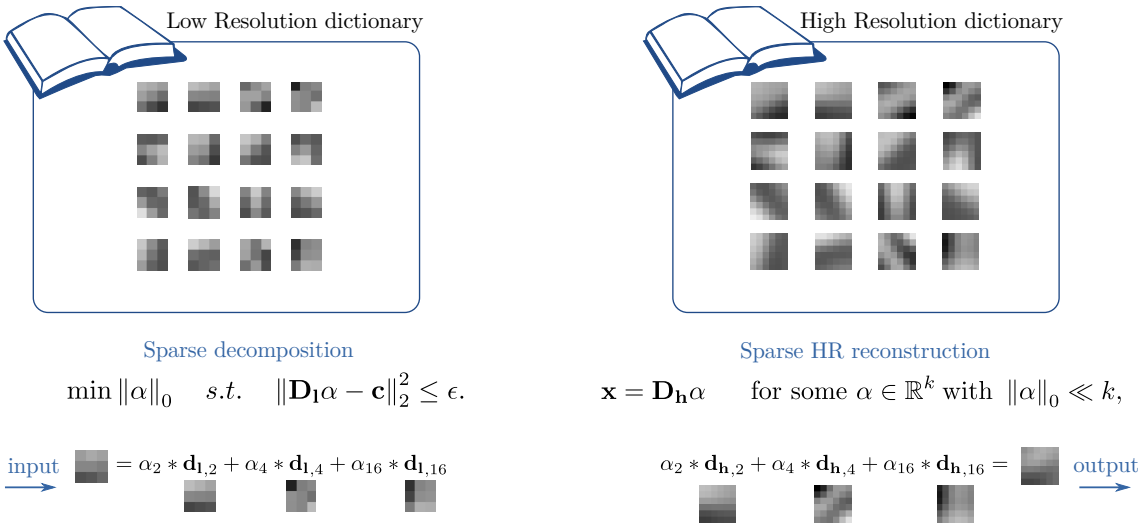


Figure 2.2: Working principle of sparse dictionary-based SR. Left side figure shows how the patch c is sparsely decomposed with respect to $\mathbf{D_l}$. Right side figure shows how the HR upscaled patch is reconstructed with α and $\mathbf{D_h}$.

As briefly noted in the introduction, the upscaling process is highly ill-posed as an infinity of potential HR images respect the reconstruction constraint of Equation (2.1). To further alleviate the problem the sparsity prior is introduced.

The core idea of sparse signal representation is that linear relationships between signals can be precisely reconstructed from their low-dimensional projections [41]:

$$\mathbf{x} = \mathbf{D_h}\alpha \qquad \text{for some } \alpha \in \mathbb{R}^k \text{ with } \|\alpha\|_0 \ll k, \tag{2.2}$$

where α is the sparse representation with reduced non-zero entries ($\|\alpha\|_0 \ll k$) and $\mathbf{D_h}$ an overcomplete dictionary containing HR patches. To recover $\mathbf{x}$, the sparse representation α will be calculated from LR patches $\mathbf{c}$ with respect to a dictionary containing the correspondent LR patches $\mathbf{D_l}$. This can be formulated as follows:

$$\min \|\alpha\|_0 \quad s.t. \quad \|\mathbf{D_l}\alpha - \mathbf{c}\|_2^2 \leq \epsilon. \tag{2.3}$$

In Figure 2.2 we show an illustrative example of the principles described by Equation (2.2) and (2.3). The optimization in Equation (2.3) is NP-hard, however Donoho [19] proposes a ℓ_1-norm relaxation in order to recovers the α coefficients:

$$\min \|\alpha\|_1 \quad s.t. \quad \|\mathbf{D_l}\alpha - \mathbf{c}\|_2^2 \leq \epsilon, \tag{2.4}$$

which the following equivalent formulation for Lagrange multipliers:

$$\min_{\alpha} \|\mathbf{D_l}\alpha - \mathbf{c}\|_2^2 + \lambda \|\alpha\|_1, \tag{2.5}$$

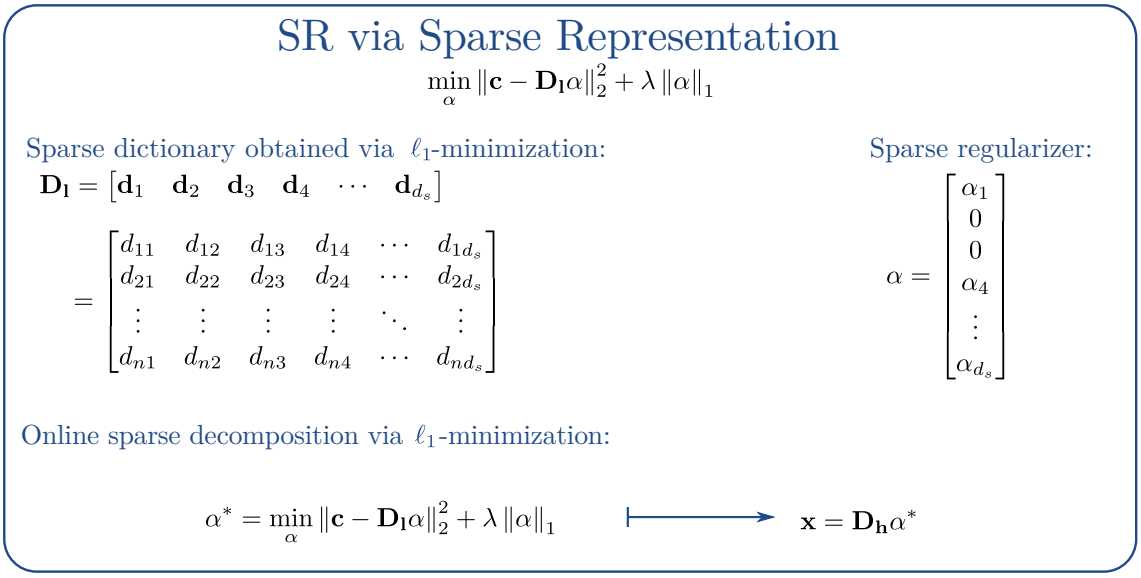


Figure 2.3: Overview of the sparse SR of Yang et al. [95] and the dimensionality of the matrices involved. Firstly, the sparse representation α is obtained through a minimization function with ℓ_1 regularization. Secondly, the sparse weight vector is applied within the HR dictionary to obtain the HR patch $\mathbf{x}$.

where λ weights the sparsity of the solution against the first fidelity term. We show an overview of the sparse SR approach of Yang et al. [95] in Figure 2.3. It is important to note that this is essentially a linear regression with ℓ_1-norm regularization. We explore other regularization options, and their effects in the upcoming chapters.

In order to guarantee a certain compatibility between the current processed patch and the previously reconstructed patches, a high-resolution fidelity term can be added in the overlap areas:

$$\min_{\alpha} \left\| \tilde{\mathbf{D}}\alpha - \tilde{\mathbf{c}} \right\|_2^2 + \lambda \|\alpha\|_1, \tag{2.6}$$

where $\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D_l} \\ O\mathbf{D_h} \end{bmatrix}$ and $\tilde{\mathbf{c}} = \begin{bmatrix} \mathbf{c} \\ \mathbf{w} \end{bmatrix}$, O is a mask describing the overlap between the current patch and the previously computed patches, w denotes the previously calculated patches in the area of overlap.

Once the optimal solution α^* to Equation (2.6) is obtained, the HR patch can be reconstructed as the linear composition $\mathbf{x} = \mathbf{D_h}\alpha^*$.

2.3 Global Reconstruction Constrain

In the previous section we studied a sparse model for image reconstruction, however the equations presented do not enforce coherency between the LR image and the

obtained upscaled image in terms of the generation model in Equation (2.1). In order to correct potential deviations from the observed LR image, we project back the image obtained through sparse reconstruction X_s into the solution space $Y = S \downarrow (H * X)$:

$$X^* = \arg\min_X \|S \downarrow (H * X) - Y\|_2^2 + \|X - X_s\|_2^2 , \tag{2.7}$$

where the second term enforces a solution which is closer to the initial SR estimated image X_s. The kernel H is normally assumed to be low-pass bicubic filter, even though it could be estimated as well. If addressed through gradient descent, the update equation reads:

$$X_{t+1} = X_t + \nu(S \uparrow (Y - S \downarrow (H * X)) + (X - X_s)), \tag{2.8}$$

where X_t represents the upscaled image after the t-th iteration, and ν is the step size of the gradient descent. The convergence is fast and there is little or no change after a reduced number of iterations. It is, therefore, a convenient and efficient post-processing stage. However, we also propose the usage of this algorithm as a first coarse estimator (i.e. as a preprocessing step) with promising results (see Chapter 6).

2.4 Training coupled dictionaries

The sparse prior decomposes the input patches with respect to two coupled LR and HR dictionaries. An straightforward approach to obtain the two coupled dictionaries is to directly sample them from training images, thus taking advantage of the already present LR-to-HR correspondence. This approach coincides with the initial approximation of Yang et al. [94] to sparse SR, where they used extensive raw patch pairs directly extracted from images. The main problem of such approach is that good generalization capabilities come at the cost of very large dictionaries and results in prohibitive computational cost when solving the minimization in Equation (2.6).

Sparse coding has tackled this problem by proposing algorithms that create compact, overcomplete dictionaries suitable for sparse reconstruction. The information contained in them is, therefore, more representative than direct raw samples and has better generalization capabilities given the same dictionary size. A well-known formulation to obtain such dictionaries requires minimizing over the sparse codes but also over the dictionary itself [41, 51, 49] :

$$\mathbf{D} = \arg\min_{\mathbf{D},\, A} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_2^2 + \lambda \|A\|_1 \;\; s.t. \;\; \|\mathbf{d}_i\|_2^2 \leq 1, i = 1, 2, \ldots, k. \tag{2.9}$$

The ℓ_2-norm fidelity term enforces an optimal dictionary, while the ℓ_1-norm term enforces sparsity (as in Equation (2.6)). The constrains in the columns of $\mathbf{D}$ remove

the scaling ambiguity necessary for a correct cost minimization. Although this minimization is not convex in both $\mathbf{D}$ and $\mathbf{A}$ (i.e. matrix in which each column is a different α), it is convex if they are addressed separately (i.e. is convex in one while the other one is fixed). In [95], Yang et al. proposed to proceed with an alternating minimization:

1. Initialize $\mathbf{D}$ with a Gaussian normalized matrix.
2. Fix $\mathbf{D}$, update A as follows (via linear programming):

$$\mathbf{A} = \arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_2^2 + \lambda \|A\|_1 . \tag{2.10}$$

3. Fix A, update as follows:

$$\mathbf{D} = \arg\min_{\mathbf{D}} \|X - \mathbf{D}\mathbf{A}\|_2^2 \ s.t. \ \|\mathbf{D}_i\|_2^2 \leq 1, i = 1, 2, \ldots, k, \tag{2.11}$$

   as it does no longer have the sparsity ℓ_1-norm term, it can be solved by many optimization packages, e.g. Quadratically Constrained Quadratic Programming [57].

4. Repeat steps 2. and 3. until convergence.

In the specific case of SR sparse dictionary learning, there are two classes of dictionaries to be learnt: LR and HR. The approach that Yang et al. proposes consists in a coupled learning, where the cost functions includes both the cost of LR reconstruction and HR reconstruction, so that the dictionaries can be different, but share the same codes:

$$\min_{\mathbf{D_h}, \mathbf{D_l}, A} \frac{1}{N} \|\mathbf{X} - \mathbf{D_h}\mathbf{A}\|_2^2 + \frac{1}{M} \|\mathbf{Y} - \mathbf{D_l}\mathbf{A}\|_2^2 + \lambda(\frac{1}{N} + \frac{1}{M}) \|A\|_1 . \tag{2.12}$$

This training procedure that allows different but coupled dictionaries implies an increase of complexity, as we need to minimize over double the dimensionality when compared to a single dictionary.

2.5 Efficient sparse SR

Some efforts have been done in order to alleviate the aforementioned limitations, specially in terms of computational times. The work of Zeyde et al. [97] is remarkable as it represents a mature SR method based on sparsity, in which efforts to optimize the whole pipeline synergize with accuracy of the upscaled images. In terms of computational complexity, Zeyde et al. [97] proposes mainly three modifications to the

original Yang et al. method [95]: (1) It obtains the LR sparse dictionary through k-Singular Value Decomposition (SVD), (2) obtains the HR dictionary $\mathbf{D_h}$ by utilizing the same encoding as in $\mathbf{D_l}$ and (3) prefers Orthogonal Matching Pursuit (OMP) to ℓ_1-optimization-based methods (e.g. Least Absolute Shrinkage and Selection Operator (LASSO)). We consider important to briefly introduce the foundation of the mechanism of k-SVD and the way that Zeyde et al. use it for SR.

2.5.1 k-SVD

The problems of clustering and sparse representation are inherently related, as they both target Vector Quantization (VQ) [20, 40]. Clustering finds a set of descriptive vectors $\{\mathbf{d}_k\}_{k=1}^{K}$ that are representative of K diverse groups in such a way that any sample can be represented by one of those vectors (i.e. normally the closest in ℓ_2 distance). In sparse coding, each sample is represented by a linear combination of several vectors within the dictionary, and thus can be explained as an extension or generalization of the clustering analysis. In a similar way the k-SVD algorithm of Aharon et al. [1] builds and generalizes the original k-Means algorithm [44, 15]. In Equation (2.9) we show a formulation for dictionary optimization based on ℓ_1-norm minimization. A different approach to enforce sparsity consists in establishing a sparsity constrain T_0 of maximum number of nonzero entries:

$$\mathbf{D} = \arg\min_{\mathbf{D},\, A} \|\mathbf{Y} - \mathbf{DA}\|_F^2 + \ s.t.\ \|\mathbf{x}_i\|_0 \leq T_0. \tag{2.13}$$

Again, the minimization problem is addressed in an iterative way, alternating between the minimization over A and the minimization over $\mathbf{D}$. First, we fix $\mathbf{D}$ and obtain the optimal $\mathbf{A}$ with any approximation pursuit method that allow a solution with a fixed and predetermined number of nonzero entries, e.g. OMP [78, 63]. The second stage consists on the search of a better suited dictionary for sparse decomposition. In this stage, the approach of k-SVD differs greatly from aforementioned coupled dictionary training. This two steps are repeated until the desired stopping rule is satisfied.

In k-SVD the dictionary optimization is tackled by updating one column at a time, fixing all columns in $\mathbf{D}$ but one, $\mathbf{d}_u$. A new column $\mathbf{d}_u$ and the new related coefficients α^u (i.e. α^u is the u-th row vector in $\mathbf{A}$) are found so that they best reduce the Mean Squared Error (MSE). Note that, in other approaches, the coefficients $\mathbf{A}$ are completely fixed during minimization over $\mathbf{D}$, whereas k-SVD approach allows to modify the relevant coefficients α_u, which accelerates convergence. Updating a single column per iteration has a known solution using SVD. More formally, we can decompose the penalty term between the fixed elements and those which are going to be updated:

$$\begin{aligned}\|\mathbf{Y}-\mathbf{DA}\|_F^2 &= \left\|Y-\sum_{j=1}^{K}\mathbf{d}_j\alpha^j\right\|_F^2 \\ &\left\|\left(Y-\sum_{j\neq u}\mathbf{d}_j\alpha^j\right)-\mathbf{d}_u\alpha^u\right\|_F^2 \\ &\|\mathbf{E}_u-\mathbf{d}_u\alpha^u\|_F^2 \,. \end{aligned} \tag{2.14}$$

The matrix $\mathbf{E}_u$ contains the error accumulated by all training samples when the u-th sample is removed. If we only change one column at a time, $\mathbf{E}_u$ is therefore fixed and only $\mathbf{d}_u\alpha^u$ remains in question. We could find a suitable solution that minimizes the reconstruction error by means of SVD, however this solution is not likely to be sparse, as we do not enforce the sparsity constrain in any way.

In order to do so, Aharon et al. [1] propose an intuitive solution which consists on enforcing always the same representation support, i.e. the examples that make use of the new u-th dictionary atom stay constant so that the maximum sparsity T_0 selected during the previous approximation pursuit stage is respected. Let us define a group of indices describing which examples $\{\mathbf{y}_i\}$ use the atom $\mathbf{d}_u$:

$$\omega_u = \{i \mid 1 \leq i \leq K,\, \alpha^u(i) \neq 0\}\,, \tag{2.15}$$

from which we can derive a matrix $\mathbf{\Omega}_u$ with ones on the $(\omega_u(i),i)$th entries and zeros elsewhere. This matrix can be seen as a shrinking operator, as the multiplication $\alpha_R^u = \alpha^u\mathbf{\Omega}_u$ discards all the zero entries and thus leave only the elements of the training set that are affected by a modification in $\mathbf{d}_u$. We can apply this shrinking multiplication to the error $\mathbf{E_u^R} = \mathbf{E}_u\mathbf{\Omega_u}$, selecting thus only the error columns that are relevant to a change in atom $\mathbf{d}_u$. Now we can force the solution to have constant support by including $\mathbf{\Omega}_u$ in Equation (2.14):

$$\|\mathbf{E}_u\mathbf{\Omega}_u-\mathbf{d}_u\alpha^u\mathbf{\Omega}_u\|_F^2 = \left\|\mathbf{E}_u^R=\mathbf{d}_u\alpha_R^u\right\|_F^2 \,. \tag{2.16}$$

SVD decomposes it to $\mathbf{E}_u^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^\intercal$, where the solution $\mathbf{d}_u$ is approximated by the first column of $\mathbf{U}$, and the solution for the weights α_R^u by the first column of $\mathbf{V}$ multiplied by $\mathbf{\Delta}(1,1)$.

The k-SVD algorithm repeat the previous SVD decomposition for each of the columns within the dictionary, always incorporating the modifications of the previous steps. Once the algorithm updated all the columns of the dictionary, the codebook stage is over. The sparse coding stage and the codebook stage are repeated until the stopping criteria is met.

As for SR, Zeyde et al. used a de-coupled dictionary training based only on LR training data. The $\mathbf{D_l}$ is obtained through Equation (2.13), and then the HR dictionary is obtained with the same LR sparse codes:

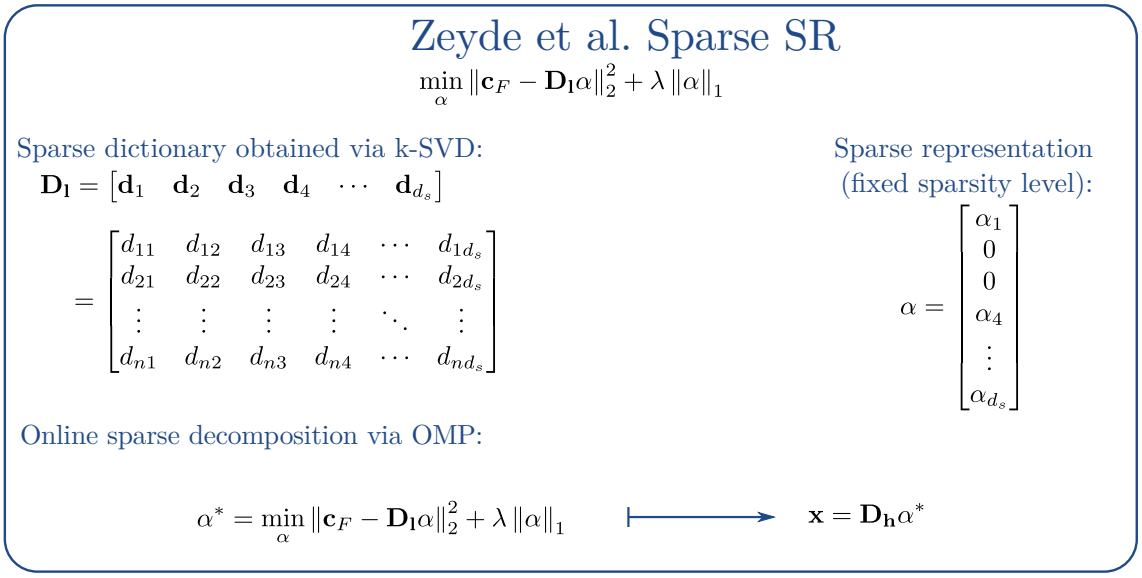


Figure 2.4: Overview of Zeyde et al. Sparse SR. The functioning is similar to the algorithm presented by Yang et al. [95], however OMP methods and k-SVD are preferred over other iterative optimization methods.

$$\mathbf{D_h} = \mathbf{X}\mathbf{A}^{-1}, \tag{2.17}$$

which can be calculated in a more memory efficient way as $\mathbf{D_h} = \mathbf{X}\mathbf{A}^\intercal(\mathbf{A}\mathbf{A}^\intercal)^{-1}$. We show an overview of Zeyde et al. SR in Figure (2.4). The combination of k-SVD together with Equation (2.17) is substantially faster than the previous approach presented in Section 2.4, and has been widely adopted by several follow-ups, including the work presented in this thesis.

2.6 Summary and discussion

In the work on sparse SR of Yang et al. [94, 95] they firstly proposed a SR model based on the sparsity prior which functions with a set of two coupled dictionaries, and also a procedure on how to train them. They proposed as well a reconstruction constrain or generation constrain that has been widely adopted. Their algorithm outperforms and is notably faster than some of its *learning-based* predecessors [25, 11]. Despite that, the computational complexity is still high, mainly due to the iterative minimizations that ℓ_1 regularization terms require: Training the coupled dictionaries takes several hours, and the upscaling of a single frame takes several minutes. As a follow-up, Zeyde et al. [97] designed a sparse model that substitutes some of the bottlenecks in terms of computational time for more efficient approaches, e.g. k-SVD or OMP, together with an uncoupled dictionary training scheme. This

results in training times of less than an hour, and testing times in the order of seconds per frame.

Both algorithms are fundamental to the understanding of this dissertation as they lay the ground and the terminology on which we build our contributions.

Chapter 3

Anchored Neighborhood Regression

3.1 Introduction

Some of the most defining limitations of the sparse SR family of methods are related to the high computational cost inherent to the sparsity constrain.

In the sparse SR of [95] complex optimization schemes have to be adopted in order to jointly obtain $\mathbf{D_h}$ and $\mathbf{D_l}$, resulting in great computational cost. During testing time, they minimize the following function:

$$\min_{\boldsymbol{\alpha}} \|\mathbf{c} - \mathbf{D_l}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 , \tag{3.1}$$

where the first term ensures a good LR reconstruction and the ℓ_1-norm regularization term enforces sparsity in the solution. The sparse decomposition $\boldsymbol{\alpha}$ is then applied to $\mathbf{D_h}$ to obtain the HR patch. This decomposition is computed for all the patches $\mathbf{C}$ in the image. Due to the ℓ_1-norm regularization term, there is no closed-form solution for such minimization, and thus they require the usage of expensive iterative procedures.

Later work on sparse SR by Zeyde et al. [97] introduced faster algorithms for dictionary optimization (e.g. k-SVD [1]) and a different optimization scheme: the dictionaries are learned separately, obtaining first $\mathbf{D_l}$ independently from $\mathbf{D_h}$, and afterwards the latter is generated with the sparse encoding of $\mathbf{D_l}$. The execution time is improved with respect to the original work of Yang et al. [95].

Despite alleviating some of the most time-consuming processing of its predecessor, the sparse decomposition in [97] is still the bottleneck during inference time. As a natural solution for that, Timofte et al. proposed the Anchored Neighborhood Regression (ANR) [74] where there is no sparse decomposition during inference time, but instead a selection within a discrete set of points (i.e. anchor points) for which a linear ridge regressor has been trained off-line. This method coincided with other similar regression-based SR algorithms such as [92, 93], and also triggered several follow-ups, e.g. [62, 76, 68]. The motivation and functioning of the ANR SR algo-

rithms is fundamental to the understanding of the contributions that conform this thesis.

3.2 Collaborative Norm Relaxation

As discussed previously, Equation (3.1) does not have a closed-form solution and requires iterative algorithm in order to find meaningful minima. Timofte et al. proposed a relaxation of the ℓ_1-norm regularization commonly used in most of the neighbor embedding and sparse coding approaches, reformulating the problem as a least squares ℓ_2-norm regularized minimization, also known as Ridge Regression.

While solving ℓ_1-norm constrained minimization problems is computationally demanding, when relaxing it to a ℓ_2-norm, a closed-form solution can be used. Their proposed minimization problem reads

$$\min_{\beta} \|\mathbf{c}_F - \mathbf{D_l}\beta\|_2^2 + \lambda \|\beta\|_2 , \tag{3.2}$$

where $\mathbf{c}_F$ is a feature extracted from a interpolated image C. The algebraic solution is

$$\beta = (\mathbf{D_l}^\intercal \mathbf{D_l} + \lambda \mathbf{I})^{-1} \mathbf{D_l}^\intercal \mathbf{c}_F. \tag{3.3}$$

The coefficients of vector β are applied to the corresponding HR dictionary $\mathbf{D_h}$ to reconstruct the HR patch, i.e. $x = \mathbf{D_h}\beta$. This can also be written as the matrix multiplication $\mathbf{x} = R\,\mathbf{c}_F$, where the projection matrix (i.e. regressor R) reads:

$$R = \mathbf{D_h}(\mathbf{D_l}^\intercal \mathbf{D_l} + \lambda \mathbf{I})^{-1} \mathbf{D_l}^\intercal, \tag{3.4}$$

and can be calculated off-line during training time.

The advantage of this norm relaxation is that a closed-form solution exists, and additionally, it can be computed off-line. This reduce the testing time greatly as only a matrix multiplication needs to be performed for each input patch $\mathbf{c}_F$ with a single unique regressor R that the authors name as Global Regressor (GR). However, this approach shows very little adaptation to the input patches, as R stays always the same regardless of the diversity of input patches.

In order to improve a finer linearization of the regression function, Timofte et al. [74] also propose using a non-fixed dictionary support, namely neighborhood, allowing different subsets of pair examples for each minimization.

3.3 Neighborhood Embedding

In order to extend the global regressor R to a set of regressors $\{R_i\}$, Timofte et al. [74] introduced a modification to Equation (3.2) by substituting $\mathbf{D_l}$ by a different

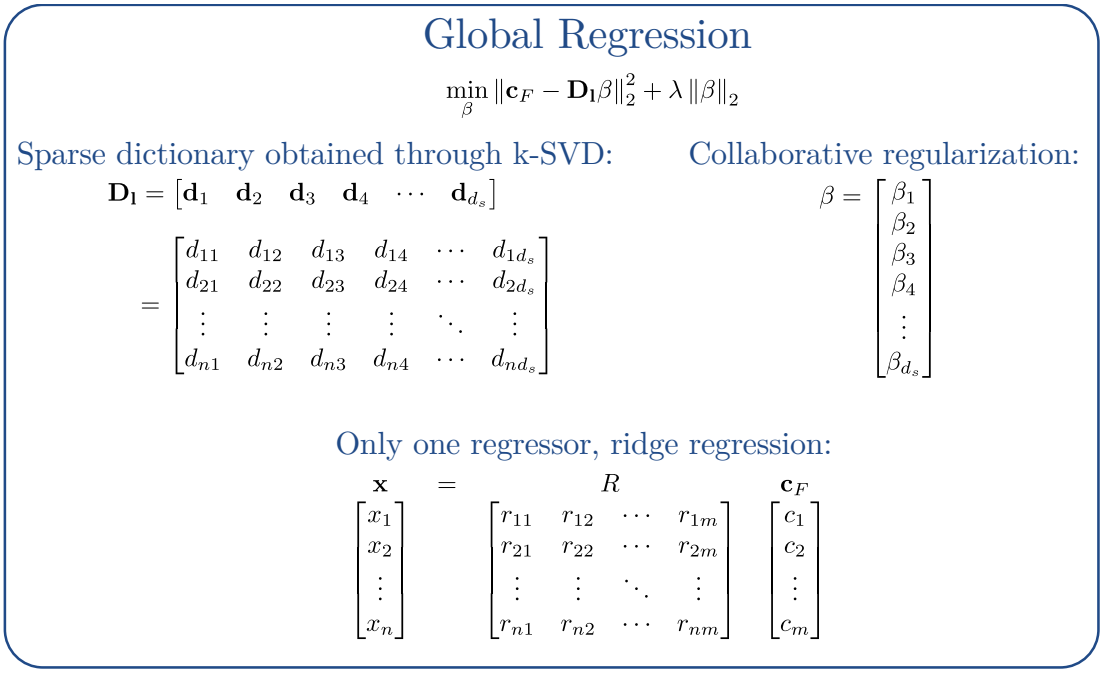


Figure 3.1: Overview of the GR matrices dimensionality.

dictionary support, namely the neighborhood $\mathbf{N_l}$:

$$\min_{\beta} \|\mathbf{c}_F - \mathbf{N_l}\beta\|_2^2 + \lambda \|\beta\|_2^2 , \tag{3.5}$$

where the neighborhood $\mathbf{N_l}$ is a subset of the original sparse dictionary $\mathbf{D_l}$(i.e. $\mathbf{N_l} \subseteq \mathbf{D_l}$), and is constructed based on the distance of the input patch or features $\mathbf{c}_F$ to each of the dictionary atoms, more formally:

$$\mathbf{N_l} = k\text{NN}(\mathbf{c}, \mathbf{D_l}) = \arg\min_{\mathbf{d_i} \in \mathbf{D_l}} \sum_{k} \delta(\mathbf{c}, \mathbf{d}_i), \tag{3.6}$$

where $\mathbf{N_l}$ columns contain the k-Nearest Neighbor (NN) atoms of $\mathbf{c}$ against the dataset $\mathbf{D_l}$ with respect to a distance measure δ.

During testing time we should first obtain the corresponding neighborhood with Equation (3.6), and then use it to find the minimum to Equation (3.5) as follows:

$$\mathbf{x} = \mathbf{N_h}(\mathbf{N_l}^\intercal \mathbf{N_l} + \lambda \mathbf{I})^{-1} \mathbf{N_l}^\intercal \mathbf{c}_F \tag{3.7}$$

which is specific to each input patch, and therefore, can not be computed offline. Although this minimization is substantially faster than solving iteratively a ℓ_1 regularized problem (see Equation (3.1)), the computation time has an impact to the overall processing, specially when using large neighborhoods.

As a natural solution to that, they propose a discretization of the testing scheme, so that instead of finding a neighborhood for each input patch, it is done during

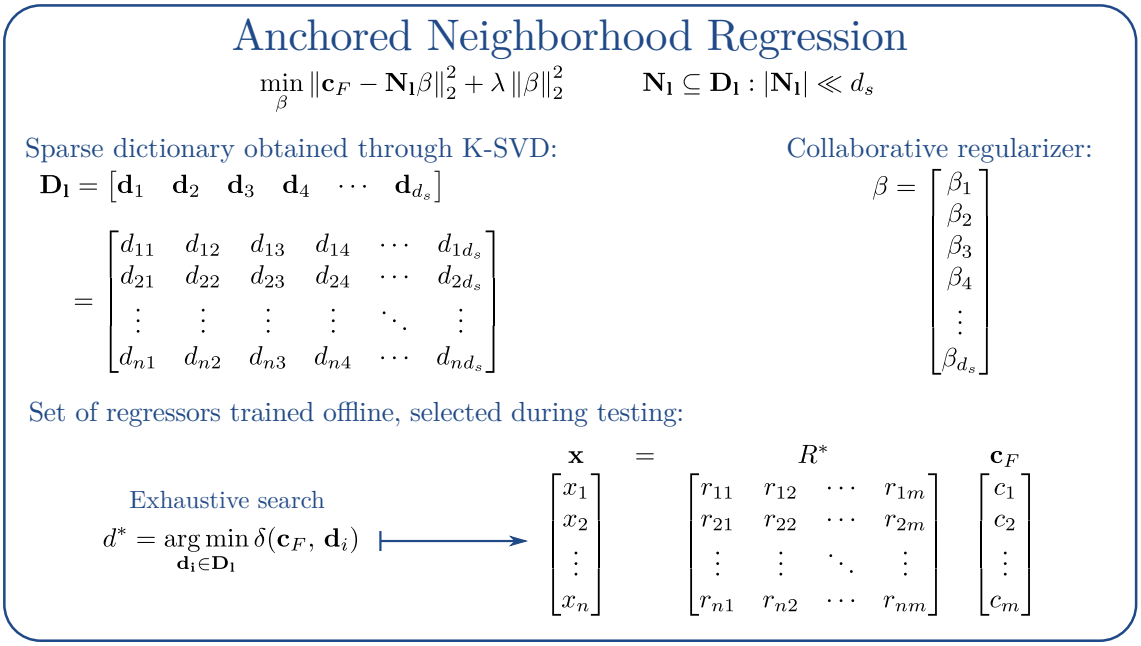


Figure 3.2: Overview of ANR.

training time for a fixed set of points, i.e. anchor points, for which a regressor is trained.

A regressor R_j is anchored to each atom $\mathbf{d}_j$ in $\mathbf{D_l}$, and the neighborhood $\mathbf{N_l}$ is selected from a k-NN subset of $\mathbf{D_l}$:

$$\mathbf{N_{l_j}} = \text{kNN}(\mathbf{d}_j, \mathbf{D_l}), \tag{3.8}$$

and later use to train a set of regressors

$$R_j = \mathbf{N_{h_j}}(\mathbf{N_{l_j}^\intercal}\mathbf{N_{l_j}} + \lambda \mathbf{I})^{-1}\mathbf{N_{l_j}^\intercal}, \tag{3.9}$$

The SR problem can be addressed by finding the NN atom $\mathbf{d}_j$ of every input patch feature $\mathbf{c}_F$ and applying the associated R_j to it. This case is referred in the original publication of Timofte et al. [74] as ANR.

3.4 Summary and discussion

The ANR and GR methods of Timofte et al. [74] introduce a ℓ_2-norm relaxation that alleviates the most significant bottleneck of previous sparse algorithms. Additionally, it introduces the idea of using neighbor embeddings within the sparse dictionaries in order to obtain ridge regressors that map LR feature points to the HR domain. By precomputing a regression function anchored to each of the dictionary atoms they avoid the patch-to-dictionary decomposition during testing time. As a result, the

execution time is greatly reduced while still obtaining competitive quality results. Although ANR is still an hybrid between sparse and neighbor embedding approaches, it represents the foundation for piece-wise linear regression-based SR.

Chapter 4

Bayesian approach to adaptive dictionaries

4.1 Introduction

In [94] the dictionary is built by randomly sampling raw patches from a large set of images regardless of the image to be recovered, hence relying on gathering sufficiently diverse patches so that they can generalize for any patch to be super-resolved. Other follow-up works [95, 45] keep using the same strategy for the training, although these raw patches are compressed in a smaller number of patches through sparse coding techniques. In the Neighbor Embedding SR work of [27], a clustering in the training set is performed based on geometrical structure of patches. The k-NN query of the input LR patch is then carried out within the closest cluster, thus showing some adaptive behavior. Nevertheless, the patches to be included in the clustering are also randomly selected out of a larger set of training patches.

Intuitively, a SR system trained with semantically similar images can adapt better and learn a more specialized dictionary, which correlates better with the content of the image. However, this is a hard problem as images usually contain a non-predictable group of different elements with their characteristic textures and edges (e.g. grass, rocks, fur, sand). In this chapter we present our work on adaptive dictionary building through Bayes theorem. We divide our training dataset into sub-image entities which we call regions, and extract descriptors in order to characterize them. The key idea is that, being these regions smaller, they have more consistent texture or edge content. For every patch to be super-resolved we find its best-fitting texture region from the training images by using the efficient local Naive Bayes Nearest Neighbor (NBNN), thus ensuring that the obtained example pairs are highly correlated with the input LR patches. Furthermore, our method is not only applicable to the original sparse SR [94], but to all other SR methods using a reduced patch pair subset from the larger training image dataset, hence including dictionary optimization processes [45].

4.2 Adaptive Training Set

The performance of sparse SR methods highly depends on the content of $\mathbf{D_h}$ and $\mathbf{D_l}$ and those in turn are determined by the contents of the training examples $\mathbf{X_t}$ and $\mathbf{C_t}$, thus being these subsets of capital importance for the whole SR process. In contrast to previous methods that build dictionaries selecting randomly patches from the training images [95, 94, 27, 45], in our approach we include a stage which adaptively selects the regions of the training images which better represent each of the input image patches without doing any manual image pre-selection.

The key idea is to extract training pair patches only from the regions likely to contain similar textures to the ones present in the image. By doing so, we can feed to the SR dictionary training algorithm (refer to Section 2) a new training set of patches $\mathbf{X_a}$ and $\mathbf{C_a}$ which are highly correlated with the content of the input image Y.

4.3 Bayesian Formulation

The problem we are addressing is that of finding a training subset $\mathbf{C_a}$ for a given test image Y adaptively. Each training image C_t is split in square regions Q of size L_Q. Given a input patch y from image Y, we find its training texture region Q. Assuming a uniform region prior over Q this can be achieved through a Maximum Likelihood (ML) decision rule:

$$\hat{Q} = \arg\max_{Q} p(Q \mid y) = \arg\max_{Q} p(y \mid Q). \tag{4.1}$$

Let $\{f\} = f_1, f_2, \ldots, f_l$ denote the descriptors extracted from patch y or its coarsely approximated version c. We use the Naive Bayes assumption, i.e. descriptors are independent, identically distributed [87]:

$$p(y \mid Q) = p(f_1, f_2, \ldots, f_l \mid Q) = \prod_{i=1}^{l} p(f_i \mid Q), \tag{4.2}$$

then, the log likehood reads:

$$Q = \arg\max_{Q} \sum_{i=1}^{l} \log p(f_i \mid Q). \tag{4.3}$$

This Maximum-a-posteriori (MAP) decision requires computing the probability density $p(f \mid Q)$, which can be obtained through a NN approximation of a Parzen density estimation $p_{NN}(f \mid Q)$ [52], as proposed by [7]. For that purpose, let then $\left\{f^Q\right\} = f_1^Q, f_2^Q, \ldots, f_L^Q$ be all the descriptors of a region Q, where f_j^Q is the jth descriptor. The Parzen kernel $K(f_i - f_j^Q) = \exp(\frac{1}{2\sigma^2} \left\| f_i - f_j^Q \right\|^2)$ yields negligible

Algorithm 4.1 AdaptiveTraining(Y,R)

Input: A Nearest Neighbor index containing all descriptors from all regions, queried by NN(d,#$neighbors$).
Input: Region lookup function Region(*descriptor*) that retrieves the region to which *descriptor* belongs to.
Input: Sampling patches function SampPatches(*Region*) which extracts patches with a certain overlap.

```
for all patches y ∈ Y do
  for all descriptors f_i ∈ y do
    {p_1,p_2,...,p_{k+1}} ← NN(f_i,k+1)
    for all regions Q found in the k nearest neighbors do
      dist_Q = min_{p_j | Region(p_j)} ||d_i − p_j||^2
    end for
    totals[Q] ← totals[Q] + dist_Q − dist_B
  end for
  Selected[y] ← arg min_Q totals[Q]
end for
for all Selected unique regions do
  T ← SampPatches(Selected[Q])
end for
return C_a
```

values for very distant descriptors since K exponentially decreases with distance. Therefore, using only the r NN of descriptor f will accurately approximate the Parzen estimation:

$$p_{NN}(f_i \mid Q) = \frac{1}{L} \sum_{j=1}^{r} K(f_i - jNN_Q(f_i)) \tag{4.4}$$

In [7] a minor decrease in performance is observed when using as little as $r = 1$ NN compared to the full Parzen window estimation, whereas this choice considerably simplifies Equation (4.3):

$$\hat{Q} = \arg\min_{Q} \sum_{i=1}^{n} \|f_i - NN_Q(f_i)\|^2. \tag{4.5}$$

Solving (4.5) requires calculating the distance from the patch to all existing regions in the training dataset. This might be computationally prohibitive since usual training sets can contain hundreds of images which translates in a number of regions in the order of thousands. Recent research in NBNN classifiers proposed local NBNN [47] which seizes this problem by only exploring the local neighborhood of each descriptor f_i. The runtime grows with the log of the number of categories rather than

linearly as in [7], which results in sensitive speed-ups for large numbers of categories (results in [47] show a $\times 100$ speed-up for 256 categories) while still outperforming the original method [7] in classification accuracy.

Let Q be some region and $\overline{Q}$ the set of all other regions. If we reformulate the NBNN updates as adjustments to the posterior log-odds, the alternative decision rule will be:

$$\hat{Q} = \arg\max_{Q} \sum_{i=1}^{n} \log \frac{P(f_i \mid Q)}{P(f_i \mid \overline{Q})} + \log \frac{P(Q)}{P(\overline{Q})} \tag{4.6}$$

Again, the prior can be dropped if assumed uniform over Q. The benefit of this alternative formulation as log-odds increments is that we can select the region posteriors which give a positive contribution on the sum in (4.6). The main contribution of local NBNN consists in (a) only using the closest member from the regions whose descriptors are within the k nearest neighbors of each f_i and (b) modeling the distance to the rest of the regions $P(f_i \mid \overline{Q})$ as the distance to the $k+1$ nearest neighbor.

After finding a region Q for every patch y, we will sample patches of size p_s with a certain overlap inside the selected regions and include them in LR and HR training sets $\mathbf{C_a}$ and $\mathbf{X_a}$, which will be used for training the sparse dictionaries and the sparse SR recovery as seen in Section 2. A summary including further implementation details can be found in Algorithm 4.1.

4.4 Rejecting Non-Informative Regions

Some regions extracted from the training images might not be useful since they do not contain high frequency (e.g. blurry unfocused backgrounds, uniform colors). In order to reject these regions, we apply a high-pass filter whose frequency cut is related to the magnification factor MF, intuitively requiring higher frequency content when a higher magnification factor is selected, according to:

$$f_c = 1 - \frac{\gamma}{MF}, \tag{4.7}$$

where γ weights the impact of the second addend. The energy per pixel is computed in the filtered region Q', defined as $E = \|Q'\|_2^2 / L_Q^2$. We reject a given region Q when its energy E is lower than a given threshold ε. Some examples of selected regions are shown in Figure 4.1.

4.5 Feature Space

We use Scale Invariant Feature Transform (SIFT) descriptors for our region selection stage. This is independent of the features that are later used by the SR algorithm

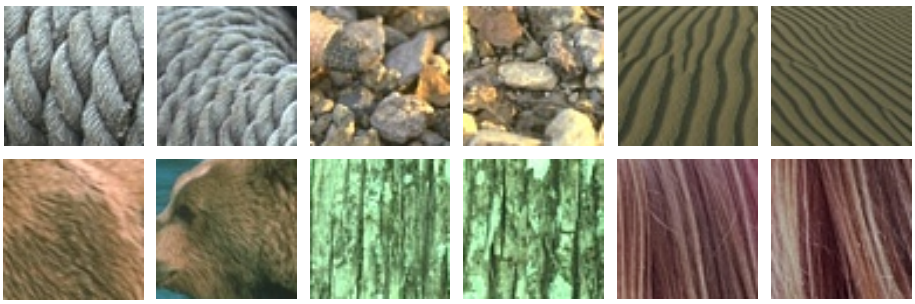


Figure 4.1: Appearance of 50x50 regions extracted from the training images. From left to right and top to bottom: rope, stones, sand, bear fur, tree bark, hair. Super-resolution performance can be improved by selecting a meaningful set of regions for every input image.

itself (e.g. concatenated gradients, mean-subtracted patches), as it only affects the region selection procedure.

SIFT descriptors show improved resilience to changes in image scale and rotation, and they are robust to changes in illumination, noise and viewpoint. They have been extensively used for detection, classification and matching across different scenes and object appearances [7, 47, 98]. We use dense SIFT extraction instead of the original SIFT detector since we are dealing with small patches and we need to force a certain number of features per patch.

4.6 Summary and discussion

We introduce a novel sparse SR method which focuses in adaptively selecting the optimal patches for the dictionary training. The method divides the training images into sub-image regions of sizes that preserve texture consistency, which are purged to reject those without high-frequency content. The best-representing region for each input LR patch is found through a Bayesian selection stage. In this selection process, SIFT descriptors are extracted densely from both input LR patches and regions and a local NBNN approach is used in order to efficiently handle the high number of different regions in the training set. The resulting adapted subset of patches is compressed using sparse coding techniques and used to recover HR images by exploiting the sparsity prior. Experimental results (we refer the reader to Chapter 9) show that our method improves performance with respect to using generic dictionaries, however it requires training new dictionaries for each new image or consistence sequence. In Chapter 7 we discuss how to use this Naive Bayes assumption as a tree selection criterion within a regression forest.

Chapter 5

Dense Local Training and Spherical Hashing

5.1 Introduction

The relaxation to ℓ_2-norm introduced by Timofte et al. in their ANR [74] differs greatly from the previous sparse ℓ_1-regularized minimization, nonetheless ANR shares some of the training practices of sparse methods as it directly inherits the same framework. ANR mimics the sparse behavior by fixing the set of atoms involved in the minimization to a relatively small, non-fixed set of neighboring atoms $\mathbf{N_l}$. Those atoms, however, are not chosen based on sparse-representation criterion, but rather based on the distance to the given anchor point (see Equation (3.6)).

In this section we propose a purely collaborative, dense training approach, moving away from the hybrid sparse neighbor embedding performed in ANR. By doing so, we obtain substantial quality gains. We also propose a sublinear search scheme that address one of the remaining most time-consuming factors during testing time: the nearest neighbor search.

5.2 Linear Regression Framework

In regression-based SR, the objective of training a given regressor R is to obtain a certain mapping function from LR to HR patches. From a more general perspective, LR patches form an input manifold M of dimension m and HR patches form a target manifold N of dimension n. Formally, for training pairs $\{\mathbf{c}_{Fi}, \mathbf{x}_i\}$ with $\mathbf{c}_F \in M$ and $\mathbf{x}_i \in N$, we would like to infer a mapping $\Psi : M \subseteq \mathbb{R}^m \rightarrow N \subseteq \mathbb{R}^n$.

Linear Regression is the most simple regression scheme, i.e. for each output variable it performs a linear weighted sum of the input variables. This is an oversimplification of the upscaling problem if we only consider one linear regressor, as the mapping Ψ is highly non linear [56]. Instead, several linear regressors are anchored to different points of the manifold, obtaining a finer piecewise linear regression model. In such strategies, data have to be split in training time and during testing time the proper regressor has to be selected.

For the SR problem, the regression is applied to the input features and aims to recover certain components of the patch, e.g. missing high frequency. We model the linear regression framework in a general way as:

$$\mathbf{x} = \mathbf{c} + R_i\, \mathbf{c}_F,\, s.t.\, R_i, \quad s.t.\ \ i = \underset{\mathbf{d}_i \in \{\mathbf{D_l}\}}{\arg\min}\, \delta(\mathbf{d}_i, \mathbf{c}_F), \tag{5.1}$$

where c is a coarse first approximation of the HR patch x, $\delta(\cdot)$ is a metric evaluating the distance from the input features to the ith regressor cluster or anchor point $\mathbf{d}_i$ with an associated regressor R_i. Note that we are explicitly recovering the residual error $(x - c)$ rather than the HR patch itself.

5.3 Neighborhoods and training

In our proposed Dense Local Training (DLT) we analyze the effect of the distribution of the regression functions in the manifold (i.e. the anchor points) and the importance of properly choosing $\mathbf{N_l}$ in Equation (3.6), concluding on a new training approach.

In the work of Timofte et al. [74], an overcomplete sparse representation is obtained from the initial LR training patches using kSVD [1]. This new reduced dictionary $\mathbf{D_l}$ is used both as anchor points to the manifold and datapoints for the regression training. In their GR, a unique regressor is trained with all elements of the dictionary, therefore accepting higher regression errors due to the single linearization of the manifold. For a more fine-tuned regression reconstruction they also propose the ANR, where they use as anchor points the dictionary atoms and they build for each one of those atoms a variable neighborhood $\mathbf{N_l}$ of k-NN within the same sparse dictionary $\mathbf{D_l}$.

Performing a sparse decomposition of a high number of patches efficiently compresses data in a much smaller dictionary, yielding atoms which are representative of the whole training dataset, i.e. the whole manifold. For this reason they are suitable to be used as anchor points, but also sub-optimal for the neighborhood embedding. They are sub-optimal since the necessary local condition for the linearity assumption is likely to be violated. Due to the ℓ_1-norm reconstruction minimization imposed in sparse dictionaries, atoms in the dictionary are not close in the Euclidean space, as shown in Figure 5.2 (a).

This observation leads us to propose a different approach when training linear regressors for SR: Using sparse representations as anchor points to the manifold, but forming the neighborhoods from a broader pool of raw manifold samples (e.g. patches or features). We propose to form $\mathbf{N_l}$ as a subset of the initial set of training patches $\mathbf{C_t}$ instead of $\mathbf{D_l}$(i.e. $\mathbf{N_l} \subseteq \mathbf{C_t}$):

$$\mathbf{N}_{\mathbf{l}_j} = \mathrm{kNN}(\mathbf{d}_j, \mathbf{C_t}), \tag{5.2}$$

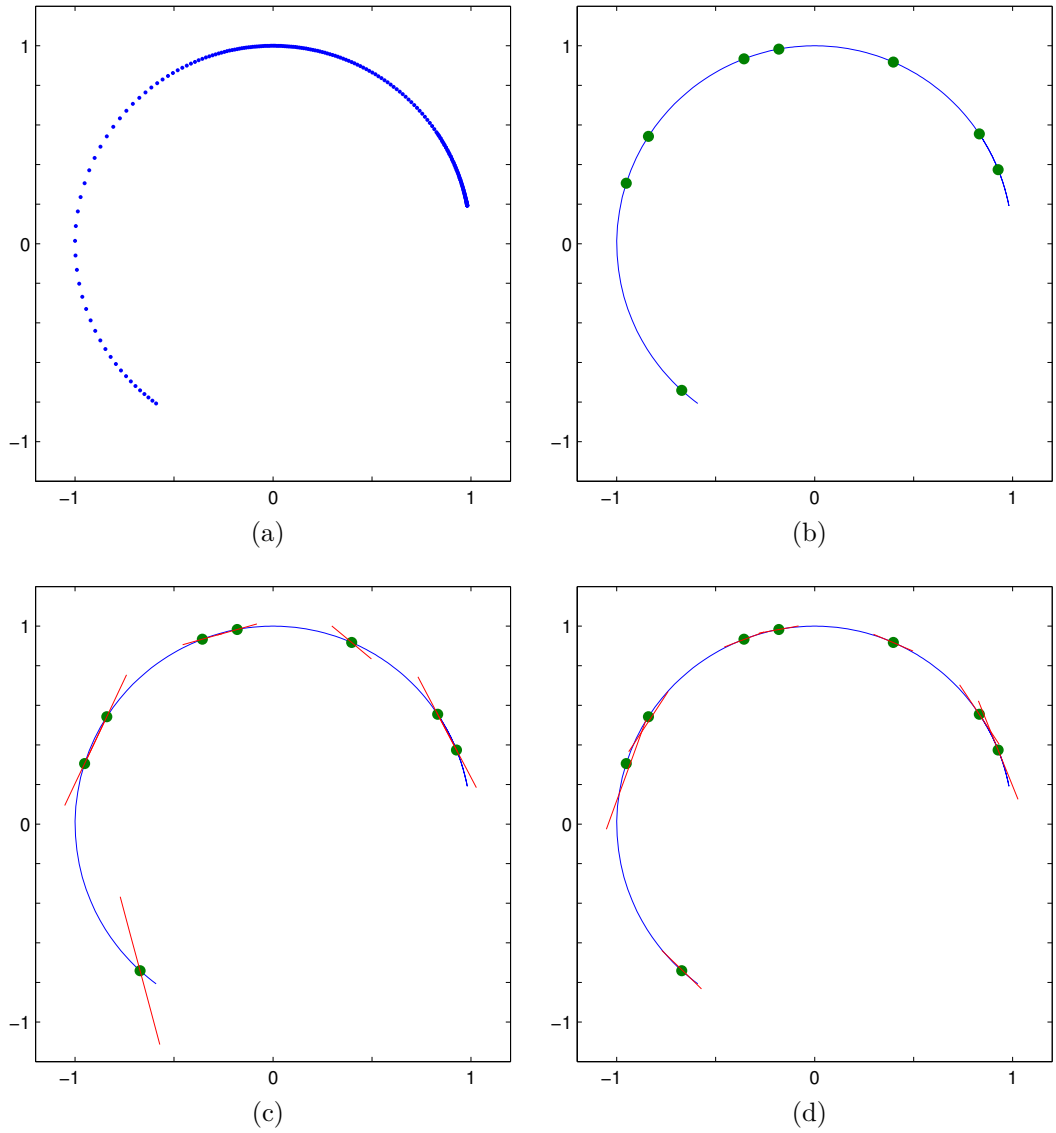
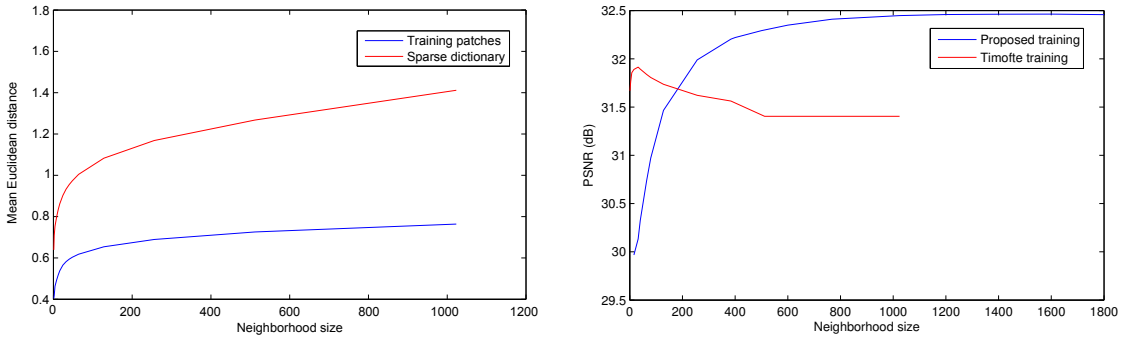


Figure 5.1: A normalized degree 3 polynomial manifold illustrating the proposed approach compared to the one in [74]. (a) Bidimensional manifold samples. (b) The manifold (blue) and the sparse representation obtained with K-SVD algorithm (green) of 8 atoms. (c) Linear regressors (red) trained with the neighborhoods ($k = 1$) obtained within the sparse dictionary, as in [74]. (d) Linear regressors (red) obtained using our proposed approach: The neighborhoods are obtained within the samples from the manifold ($k = 10$).



(a) (b)

Figure 5.2: (a) Mean euclidean distance between atoms and its neighborhood for different neighborhood sizes. (b) Quality improvement measured in PSNR (dB) for a reconstruction using ANR [74] together with our proposed training. 1024 anchor points were used for this experiment.

In Figure 5.2(a) we show how, by doing so, we find closer nearest neighbors and, therefore, fulfill better the local condition. Additionally, a higher number of local independent measurements is available (e.g. mean distance for 1000 neighbors in the raw-patch approach is comparable to a 40 atom neighborhood in the sparse approach) and we can control the number of kNN selected, i.e. it is not upper-bounded by the dictionary size. We show a low-dimensional example of our proposed training scheme in Figure 5.1. Building dense and compact clusters through this methodology is a key contribution as it boosts performance at no complexity cost. Timofte et al. also presented a similar idea in their A^+ [76] concurrent to our publication [62].

5.4 Search Strategy

When aiming at a fine modeling of the mapping between LR and HR manifolds, several linear regressors are trained to better represent the non-linear problem. Although state-of-the art regression-based SR has already pushed forward the computational speed with regard to other dictionary-based SR [97, 95], finding the right regressor for each patch is still consuming most of the execution time. In the work of [74], most of the *encoding time* (i.e. time left after subtracting shared processing time, including bicubic interpolations, patch extractions, etc.) is spent in this task (i.e. $\sim 96\%$ of the time).

We propose a novel search strategy designed to benefit from the training outcome presented in Section 5.3, i.e. anchor points of the dictionary and their neighborhoods are obtained independently and ahead from the search structure.

In order to improve the search efficiency, search structures of sublinear complexity

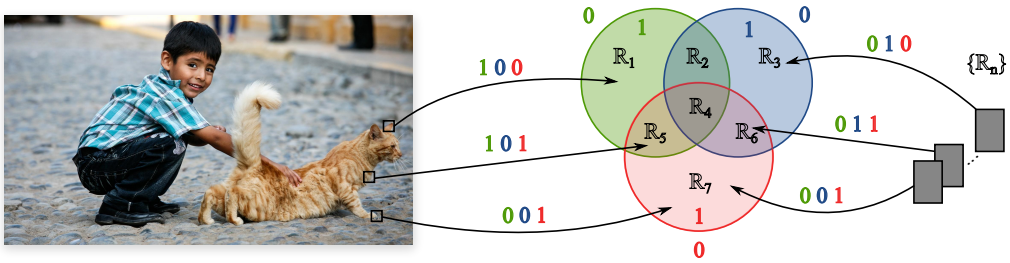


Figure 5.3: Spherical hashing applied for the anchor point search of our algorithm. Certain hashing functions are optimized on feature patch statistics creating a set of hyperspheres intersections that are directly labeled with a hash code. In training time, anchor points fill this intersections (i.e. bins) and in testing time the hashing function is applied to each patch, which will directly map it to a regressor.

are often built, usually in the form of binary splits, e.g. trees, hashing schemes [83, 36, 31, 10]. One might consider determining the search partitions with the set of anchor points, since those are the elements to retrieve. However, the small cardinality of this set leads to an imprecise partitioning due to a shortage of sampling density.

We propose to train our Spherical Hashing (SpH) functions with natural-image patches and later on label both anchor points and input patches, as we show in Figure 5.3. By doing so, we have a dense sampling (i.e. all training patches) at our disposal, which results in meaningful partitions.

Hashing schemes provide low memory usage (the number of splitting functions in hashing-based structures is $\mathcal{O}(\log_2(d_s))$ while in tree-based structures is $\mathcal{O}(d_s)$, where d_s represents the number of clusters) and are highly parallelizable.

Binary hashing techniques aim to embed high-dimensional points in binary codes, providing a compact representation of high-dimensional data. Among their vast range of applications, they can be used for efficient similarity search, including approximate nearest neighbor retrieval, since hashing codes preserve relative distances. There has recently been active research in data-dependent hashing functions opposed to hashing methods such as [36] which are data-independent. Data-dependent methods intend to better fit the hashing function to the data distribution [88, 83] through an off-line training stage.

Among the data-dependent state-of-the-arts methods, we select the Spherical Hashing algorithm of Heo et al. [31], which is able to define closed regions in $\mathbb{R}^m$ with as few as one splitting function.

Spherical hashing differs from previous approaches by setting hyperspheres to define hashing functions on behalf of the previously used hyperplanes. A given hashing function $SH(c_F) = (h_1(c_F), \ldots, h_c(c_F))$ maps points from $\mathbb{R}^m$ to a base 2 $\mathbb{N}^s$, i.e. $\{0,1\}^s$. Every hashing function $sh_k(c_F)$ indicates whether the point c_F is inside kth hypersphere, modeled for this purpose as a *pivot* $p_k \in \mathbb{R}^m$ and a distance

threshold (i.e. radius of the hypersphere) $t_k \in \mathbb{R}^+$ as:

$$sh_k(c_F) = \begin{cases} 0 & when\ \delta(p_k,c_F) > t_k \\ 1 & when\ \delta(p_k,c_F) \leq t_k \end{cases}, \tag{5.3}$$

where $\delta(p_k,y_F)$ denotes a distance metric between two points in $\mathbb{R}^m$ (e.g. Euclidean distance). The advantages of using hyperspheres instead of hyperplanes is the ability to define closed tighter sub-spaces in $\mathbb{R}^m$ as intersection of hyperspheres. An iterative optimization training process is proposed in [31] to obtain the set $\{p_k,t_k\}$, aiming a balanced partitioning of the training data and independence between hashing functions.

We perform this mentioned iterative hashing-function optimization in a set of input patch features from training images, so that $SH(c_F)$ adapts to the natural image distribution in the feature space. Our proposed spherical hashing search scheme becomes symmetrical as we can see in Figure 5.3, i.e. both image and anchor points have to be labeled with binary codes. This can be intuitively understood as creating NN subspace groups (normally referred as *bins*), which we label with a regressor by applying the same hashing functions to the anchor points. Relating a hash code with a regressor can be done during training time.

This search returns kNN for each anchor point, thus it does not ensure that all the input image patches have a related regressor (i.e. whenever the patch is not within the kNN of any of the anchor points). Two solutions are proposed: (a) use a general regressor for the patches which are not in the kNN of any anchor point or (b) use the regressor of the closest labeled hash code calculated with the spherical Hamming distance, defined by [31] as $d_{SH}(a,b) = \frac{\sum(a \oplus b)}{\sum(a \wedge b)}$, where $\oplus$ is the XOR bit operation and $\wedge$ is the AND bit operation. Note that although not being guaranteed, it rarely happens that a patch is not within any of the kNN regressors (e.g. for the selected parameter of 6 hyperspheres it never occurs).

In a similar way, this search might also assign two or more regressors to a single patch. It is common in the literature to do a re-ranking strategy to deal with this issue [30].

With our proposed search strategy, the complexity of the search is $\mathcal{O}(\log_2(d_s))$ where d_s is the number of atoms in the sparse dictionary $\mathbf{D_l}$. With previous exhaustive search approaches, the complexity grow linearly as in $\mathcal{O}(d_s)$.

5.5 Summary and discussion

In this chapter we introduced a novel scheme to train regressors based on pure collaborative neighbor embedding, which fits better the ℓ_2-norm regularization behaviour. For each atom in the sparse dictionary we create a densely populated neighborhood from an extensive training set of raw patches (i.e. in the order of hundreds

of thousands), thus constructing highly populated dense neighborhoods. Training regressors with those neighborhoods results in better fitted regression functions, which in turn improve greatly the performance in terms of reconstruction quality. We also propose a Spherical Hashing sublinear search strategy in order to avoid the exhaustive search necessary to find the closest anchor point to each of the input patches. By combining both contributions, DLT obtains highly competitive quality performance and faster execution times.

Chapter 6

Half-Hypersphere Confinement

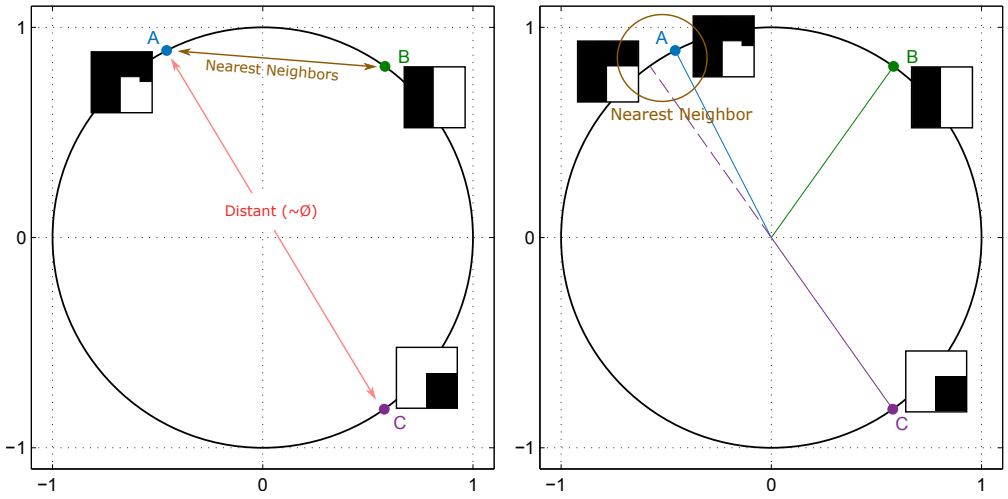
6.1 Introduction

The mapping of the manifold is assumed to be locally linear and therefore several linear regressors are used and anchored to the manifold as a piecewise linearization. The key observation of performing the neighbor embedding just for a predefined set of anchor points allowed to preprocess them during training time, thus lessening substantially the testing time complexity. Our work on dense local regression [62] puts light in how to properly create the neighborhoods in such methods, obtaining sizable benefits in terms of reconstruction quality. In addition to that, we also included sublinear search structures for the regressor nearest neighbor search, as this takes a significant quota of the running time from within the processing of the whole SR pipeline.

In this chapter we further study and provide insight about the behavior of distance metrics used during the regression process. Even though Euclidean distance has been widely used for this purpose, it is suboptimal since it does not handle antipodal points (i.e. diametrically opposite points) properly, as vectors with same module and angle but opposite directions are, for linear regression purposes, identical (see Figure 6.1). We propose the usage of antipodally invariant metrics and introduce the Half-Hypersphere Confinement (HHC), a fast alternative to Multidimensional Scaling (MDS) that allows to map antipodally invariant distances in the Euclidean space with very little approximation error. By doing so, we enable the usage of fast search structures based on Euclidean distances without undermining their speed gains with complex distance transformations.

6.2 Metrics for linear regression

The linear regression scheme is, as we have seen, very straightforward. One of the most fundamental aspects of the system is how we choose the best-suited regressor,



(a) Euclidean metric (b) Antipodally invariant metric

Figure 6.1: Behaviour of Euclidean distance and angular distance for points A, B and C. Although A and C have very similar structures, Euclidean distance fails to group them together.

i.e. the metric $\delta(= \mathbf{d}_i, \mathbf{c}_F)$ used to compare the input patch to the ith centroid in Equation (5.1).

This metric is not only important during testing time, but also during training time to asses which observations are used to train which regressor. It is recurrent in literature the use of Euclidean distance for this purpose. If we are aiming a nearest neighbor search for a regression system, Euclidean space without any transformation is suboptimal as it is not further exploiting the intrinsic characteristics of linear regression.

The scalar matrix multiplication gives us some information about the ambiguous variations that the metric we want to define should ignore, i.e. for a given scalar λ we obtain $\lambda\mathbf{x} = R(\mathbf{c}\lambda)$. The regressor R and the associated linear operations are not changed by this scaling operation. Therefore, performing a vector normalization is a good practice as it solves partially the undesired variability derived from scalar multiplication. Unitary vectors collapse all positive scalar variations into a single unitary vector, thus holding more training examples available for a certain vector type and being able to use them efficiently. During testing time, although the regression must be done with the original non-normalized vector y, the search should be done with the normalized version $\hat{\mathbf{c}} = \frac{\mathbf{y}}{\|\mathbf{y}\|}$ for the same principles.

However, there are still certain cases which are not properly managed by just a normalization, as the norms are strictly positive, i.e. $\|\mathbf{c}\| \ \epsilon \ \mathbb{R}^+$, and therefore can not compensate for all those scalar values $\lambda \ \epsilon \ \mathbb{R}^-$. In a unitary sphere composed by normalized vectors, the case of a negative λ represents its antipode (i.e. the point

that is diametrically opposed in the unitary sphere).

The antipode of a point is one of the two closest possible nearest neighbors, however in the Euclidean space they are the most far away possible points (i.e. at a diameter distance) as it can be appreciated in Figure 6.1.

Training and assigning different regressors for two antipodal points does not increase the performance by a better specialization, as the sign change is in both sides of the equality and the regressor and the associated linear operations are identical for two antipodal points (i.e. $x = R(c)$ and $-x = R(-c)$). Each regressor is associated with an anchor point, which describes a certain *mode* in the structure of patches, regardless of this structure being a positive or negative change (e.g. positive or negative change in the gradient), which is described by the sign of the normalized vector. The metric utilized for selecting the best regressor should therefore be able to associate two antipodal points to the same anchor point, thus having *antipodal invariance*. In the same way, when building the neighborhoods during training, this observation also applies. A good metric for regression should be therefore *antipodally invariant*.

We define an antipodal invariant metric as:

$$\delta(a, b) = \delta(-a, b) = \delta(a, -b) = \delta(-a, -b). \tag{6.1}$$

We propose a metric based on the Cosine Similarity (CS) as a native antipodally invariant similarity metric which is well adapted for regressors' nearest neighbor search:

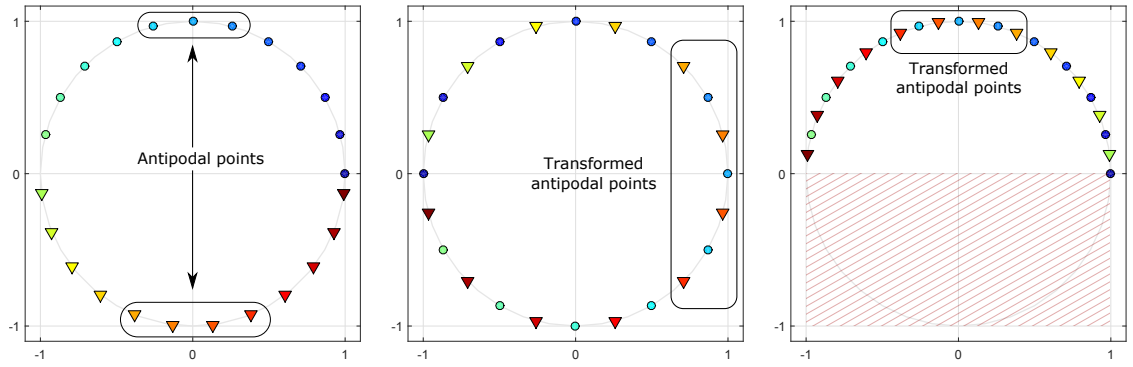
$$\varsigma(c, y) = |\hat{c} \cdot \hat{y}| = |\cos\theta|\,, \tag{6.2}$$

where the hat in $\hat{c}$ denotes unitary vectors. The output of Equation (6.2) is bounded in the [0, 1] range (1 denotes maximum similarity) and measures the absolute value of the cosine of the angle θ between the two vectors c and y. The equivalent distance metric, which we denote as angular distance reads:

$$\delta_\theta(c, y) = \frac{2}{\Pi} \arccos(\varsigma(c, y)) \tag{6.3}$$

and is normalized to be in the range [0,1] range (1 denotes maximum distance).

When there is no time nor metric space constrains (e.g. during training), using the similarity calculation of Equation (6.2) is the best option. However, during testing time if a binary split is used, and this split is making use of Euclidean space (such as the one used in [92, 62]), the adaptation is not straightforward. Rather than trying to design a split-specific metric, as in our antipodally invariant naive Bayes forest SR [65] (described in Chapter 7), we study the embedding of datapoints in the Euclidean space in such a way that antipodally invariant distances (i.e. Equation (6.3)) are preserved.



(a) Input points (b) MDS transformation (c) Proposed transformation

Figure 6.2: 2D example of our HHC compared to MDS. (a) Input points distributed on the unitary sphere (each point has its own color label), circles/triangles denote positive/negative y-axis coordinate. (b) Points obtained with MDS and angular distance. (c) Points obtained with our proposed fast transformation. To guide the reader there is a group of antipodally invariant nearest neighbors highlighted in a box across the three figures. Note how both MDS and our proposed transformed put close together antipodal nearest neighbors as opposed to the input data, where they are located at maximum distance.

6.3 Embedding in the Euclidean Space

MDS is a well known statistical method that transforms an $l \times l$ matrix D containing pairwise distances between all l observations into a set of coordinates such that the Euclidean distances derived from them preserve the relative distances specified in D. MDS is widely applied as a metric-preserving dimensionality reduction method, as the dimension of the output coordinates is user specified [58].

Although MDS can map appropriately antipodally invariant distances into Euclidean space, it is unusable as not only the optimization process is computationally intense when the number of points is very large (its complexity is $\mathcal{O}(ml^2)$, where m is the dimensionality), but additionally the input matrix D requires performing an exhaustive search point-to-point for all l elements. The Landmark MDS of Silva and Tenenbaum [13] introduced a more efficient transformation based on an approximate anchored MDS.

Silva and Tenenbaum [13] propose to divide the algorithm in two steps: a first step in which a classical MDS is performed with a smaller set of points, i.e. landmark points $l_s \ll l$, and a second step that applies a distance-based triangulation in order to obtain the embedding of the complete l elements. The first step can be done beforehand in a training stage, selecting an optimal set of landmark points, whose minimum size is $m + 1$ landmarks for a m-dimensional embedding. The embedding

vectors for each of the points can be obtained as:

$$q_a = -\frac{1}{2}(L_m^\intercal)^+(dist_a - dist_\mu), \tag{6.4}$$

where $dist_a$ is a vector with the squared distances from point a to all the landmarks, $dist_\mu$ is a vector with the mean square distances from the i-th landmark to all the landmarks (it is obtained in step 1) and $(L_m^\intercal)^+$ is the pseudoinverse transpose of L_m (it is obtained in step 1, we refer the reader to [13] for further details). Landmark MDS only requires calculating the distances for $l_s \times l$. Nevertheless, although substantially faster than the original MDS, Landmark MDS is still causing a big impact in the processing time in algorithms which have an emphasis in low complexity, such as it is our proposed work.

As a fast alternative to the MDS family of algorithms we propose a simple deterministic transformation designed to mimic the MDS behavior when it is used with angular distances. Figure 6.2 (a) and (b) show the coordinates obtained with MDS with a D matrix constructed with angular distances. If we analyze the transformation in the y-axis, the MDS transformation stretches the positive half-space points to occupy the whole sphere, and it maps likewise the negative half space. The resulting mapping contains both original half-spaces mixed in such a way that the angular distances are preserved. The intuition behind our transformation is to make use of the inverse projection ($\lambda = -1$, which is neutral for the regressor search) to compress all the data in the positive half space rather than stretching both half-spaces, as it can be seen in Figure 6.2 (c).

Several conditions need to be met for our proposed transformation to be effective. It takes advantage of the characteristics of normalized features (i.e. observations in the unitary sphere S^{m-1}). It also requires them to be distributed in both positive and negative half-spaces in a balanced way at least in one dimension.

The desired function must map two (antipodal) points in S^{m-1} into a single point. In order to do that, we enforce a forbidden space region, corresponding to the negative half-space of the qth dimension, i.e. the observations must be $c \cdot \mathbf{e_q} \in \mathbb{R}^+$, where $\mathbf{e_q}$ is the qth standard basis in the Euclidean m space:

$$\mathbf{c} = -\mathbf{c}, \qquad \text{if } \mathbf{c} \cdot \mathbf{e_q} < 0. \tag{6.5}$$

In our training (around 500K feature vectors), all the dimensions were highly and similarly balanced, so any of them could be chosen. Whenever this is not the case, the most balanced dimension should be selected to create the hyperplane.

The outcome is a HHC instead of the initial unitary hypersphere, where the Euclidean distances respect also the angular distances.

The proposed HHC is created from an hyperplane $c \cdot \mathbf{e_q} = 0$ which is the bound of the confinement. The performance of our proposed transform depends on the distance to this hyperplane, as points which are very close to the hyperplane loose

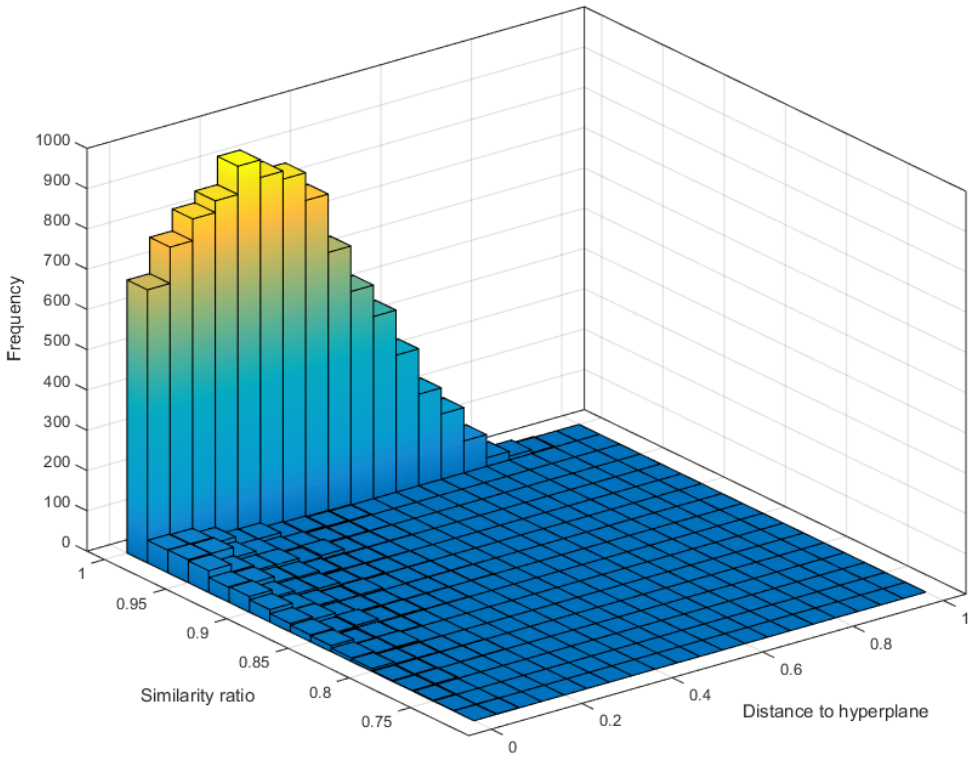


Figure 6.3: Histogram of the similarity ratio vs distance to the folding hyperplane. We evaluate the well-functioning of our proposed transformation by searching the 1-st nearest neighbor from 10k points to 1024 centroids (i.e. the testing case of our regression ensemble). We first obtain the NN both with cosine similarity (i.e. best solution) and with our proposed transformation of the Euclidean space (i.e. approximation). We recalculate the cosine similarity for both NN and compute the ratio $\eta = \zeta_{HHC}/\zeta$ (when $\eta \approx 1$ the approximation is very close to the best solution).

connection to the points immediately below the hyperplane, which are projected to the upper half hypersphere. As shown in Figure 6.2, MDS has a continuous distribution of points while our proposed transform is truncated in the $c \cdot \mathbf{e_q} = 0$ hyperplane. However, in Figure 6.3 we quantify the low incidence of this behavior by measuring a similarity ratio $\eta = \zeta_{HHC}/\zeta$ vs the distance to the hyperplane and observing its frequency. Although there is indeed a certain degradation for small distances to the hyperplane, the frequency is very low and most of the similarities obtained with HHC are highly reliable (99.2% of the total amount of points have a similarity ratio higher than 0.85).

We place a Spherical Hashing search split [31] on top of our piecewise linear regression using the HHC to embed our points in the Euclidean space. If rather than transforming the points we just want to use an antipodally invariant metric which can operate in the Euclidean space, Equation (6.5) is applied in both vectors during the Euclidean distance calculation:

$$\delta_{HHC}(p_k,y_F) = \sqrt{\sum_m (p_{k_{HHC}},c_{F_{HHC}})^2}, \tag{6.6}$$

and then used in the distance metric $\delta(p_k,y_F)$ of Equation (5.3). Thanks to this we obtain an Antipodally Invariant Spherical Hashing which is optimal for the SR regression problem and can be used for any other problem which shares the same feature characteristics.

6.4 Feature Space and coarse approximation

SR algorithms are usually performed in a feature space other than that of the raw luminance pixel values. In the literature, a common rule for this feature transformation is to enforce mid and high frequencies of LR patches, under the observation that similarity between LR and HR patch structures is somehow improved and therefore the prediction is easier. As for the HR feature space (i.e. the output feature space), the same principle of enforcing high frequencies also applies, in this case under the assumption that the high frequency bands are conditionally independent of the lower frequency bands, and thus suppressing low-frequency bands from the HR feature space collapses the training data for all possible low-frequency values into one value [24]. Differently from the input LR feature space, in the HR feature space we need to be able to reverse the features into pixel-based values for the final image reconstruction.

Several features have been proposed: The early work of Freeman et al. [24] already used a simple high-pass filter which consisted in the subtraction of a low-pass filter. In the same direction, [11] and [95] used concatenated first- and second-order gradients, as an inexpensive solution to the same high-pass filter approximation. This type of feature was further refined by Zeyde et al. [97] by introducing Principal

Component Analysis (PCA) compression in order to reduce the feature dimensionality.

It is important to remark that most feature transformations are computed from a first coarse approximation, i.e. the upscaled image C and not directly from the LR image Y. We observed that the effect of this first approximation has been unnoticed in the literature, in which using bicubic interpolation or the patch-mean value is a common practice. In this section we propose a new feature transform which takes advantage of a better coarse approximation to obtain the input features, which we denote with c_F.

The main idea is to obtain an image approximation C better than that obtained with bicubic interpolation but which is still within certain low-complexity boundaries. We present a feature transform based the Global Reconstruction Constrain of [95] (described in Section 2.3), together with unidimensional vertical and horizontal, 1-st and 2-nd order gradients. We refer to this novel feature transform as Gradient Iterative Back Projection (GIBP).

Starting with an initial guess $C^{(0)}$ of the HR image, Iterative Back Projection (IBP) simulates the imaging process to obtain a LR image $\tilde{Y}^{(0)}$ which can be compared to the observed input image Y. The difference image $E^{(0)} =\uparrow (Y - \tilde{Y}^{(0)})$ is computed and used to improve the initial guess by back-projecting each error value onto the corresponding field in $\tilde{X}^{(0)}$, namely $\tilde{X}^{(1)} = \tilde{X}^{(0)} + E^{(0)}$. This process is repeated iteratively:

$$\tilde{X}^{(n)} = \tilde{X}^{(0)} + \sum_{j=1}^{n} E^{(j-1)} \tag{6.7}$$

In our low-complexity approach, we model our downscaling and error upscaling with the simple and effective bicubic downscaling and upscaling kernel. With as few as $n = 2$ iterations the coarse approximation improves greatly when compared to bicubic. We filter this upscaled image $C^{(2)}$ with 1-st and 2-nd order unidimensional gradient filters (two vertical and two horizontal). At this point overlapping patches are extracted from each of the gradient images, and all the 4 gradient patches corresponding to the same patch position are concatenated together in a feature vector $\mathbf{c}_F$. Note that dimensionality of this feature is four times the patch size. If this eventually becomes a memory problem, a PCA compression can be applied with barely no information loss [97].

As for the HR feature space, we consistently use IBP, without the non-reversible gradient step and PCA compression. During training, we form our HR features simply by the subtraction of the the first coarse approximation to the ground-truth patch $\mathbf{x} - \mathbf{c}^{(2)}$, so that our regression stage is specialized in correcting the errors that characterize IBP. During testing, this HR feature transform requires substituting $\mathbf{c}$ by $\mathbf{c}^{(2)}$ in Equation (5.1).

Table 6.1: Average performance in terms of PSNR (dB) and time (s) for bicubic gradients features and our GIBP features, run on Set14 and Set5 on a $\times 2$ magnification factor.

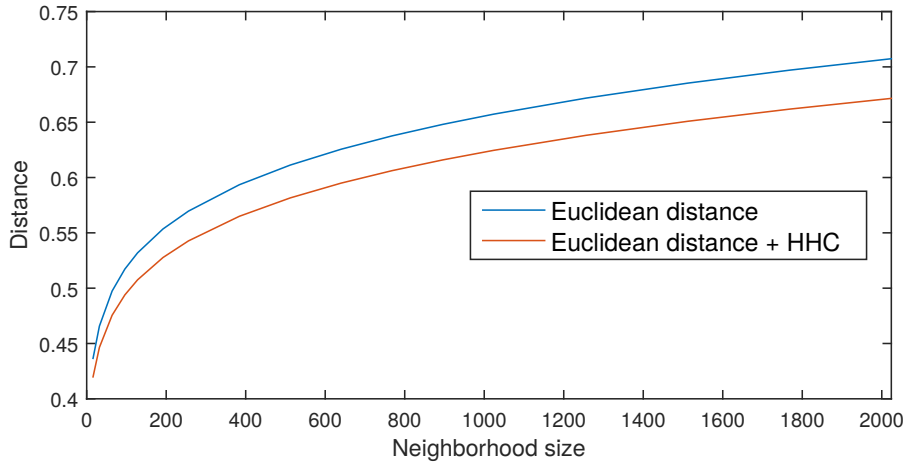
	Set5		**Set14**	
	PSNR	time	PSNR	time
bic. gradients	32.55	0.216	23.27	0.407
GIBP	32.65	0.222	32.33	0.419

6.5 Validation

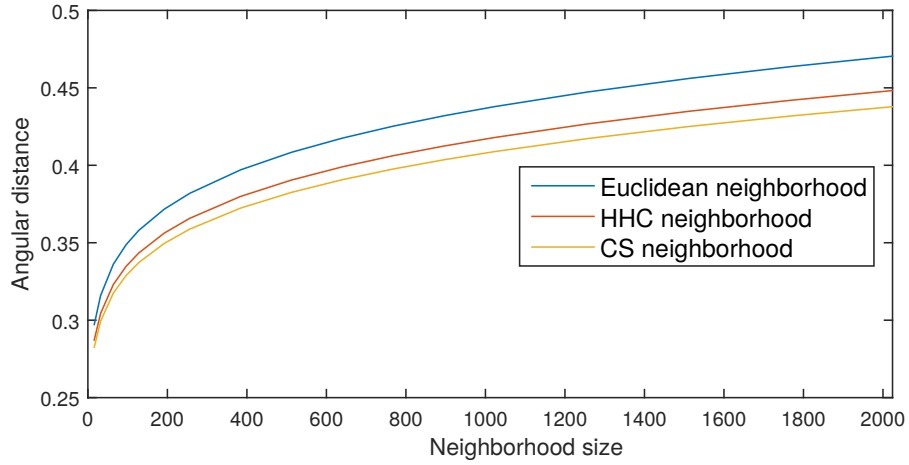
In Figure 6.4(a), we confirm that neighborhoods created after HHC metric have lower average distances than without transformation, hence obtaining a better local condition and having a higher number of samples available for a given maximum distance. In Figure 6.4(b), we asses the resilience to antipodal variance of HHC: The average angular distances obtained with HHC neighborhoods (Equation (6.6)) are approaching those created with a pure antipodally invariant metric (i.e. CS). This is further validated by the results shown in Table 6.2, where HHC and CS obtain similar PSNR performance.

In Figure 6.5 we show how the improvements of antipodality affect with respect the dictionary size and the features utilized. We show the good performance of HHC applied within SR upscaling, which approximates closely the performance obtained with CS, specially from 128 atoms on. Note that in our previous DLT SR [62], when placing a sublinear search structure based on the Euclidean distance we introduced a substantial quality drop with respect to exhaustive search. Differently, in our proposed scheme of SpH together with HHC, the drop in quality is very reduced or even nonexistent (see Chapter 9).

We compare the performance of our proposed GIBP features with the features proposed by Zeyde et al. [97] which are based on the gradients of the bicubic interpolation. We compress both features with PCA in order to reduce the dimensionality, so that not only the features are more compact, but specially the regressors. In Table 6.1 we show how by using our proposed features we consistently improve in quality (i.e. from 0.06dB to 0.10dB) with respect the previously used features. We also asses that, as expected, the computation time of the whole SR algorithm increases as GIBP requires the computation of more bicubic interpolations (three interpolation against a single one). Nevertheless the increase in running time has low incidence with respect the whole SR pipeline (i.e. about 3% of the total SR time).



(a)



(b)

Figure 6.4: Average distance of the neighborhoods to their anchor points for increasing neighborhood sizes. (a) shows the Euclidean distance of neighborhoods created before and afterHHC of the features and (b) shows angular distance (i.e. the distance derived from cosine similarity) for the neighborhoods obtained with different metrics: Euclidean distance, Euclidean distance after HHC and CS. In (a) we show how thanks to our HHC the clusters are tighter in the Euclidean space. In (b) we assess how we improve the invariance to antipodality with respect to Euclidean distance, being very close to the curve obtained with a pure antipodally invariant metric (i.e. CS).

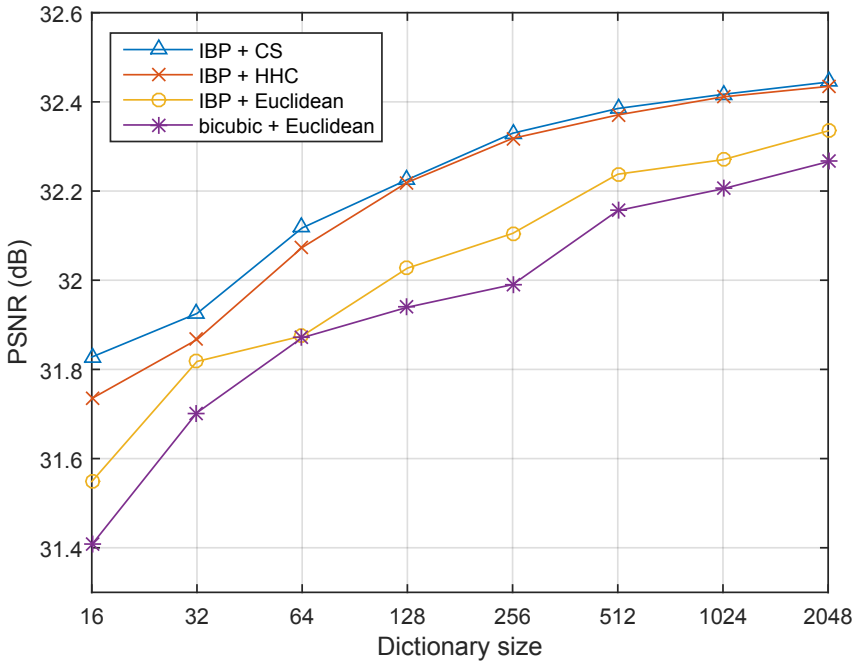


Figure 6.5: Super-Resolution upscaling Peak Signal-to-Noise Ratio (PSNR) vs dictionary size for different metrics and coarse approximations. All the configurations use exhaustive search.

Table 6.2: Performance of different metrics for training and testing run on Set14 and $\times 2$ magnification factor.

		Testing		
		Cosine	Euclidean	HHC
Training	Cosine	32.33	32.21	32.33
	Euclidean	32.27	32.15	32.26

6.6 Summary and discussion

We follow-up the DLT upscaler by studying how to improve the metrics involved in the regression nearest neighbor search both during testing and training. We detect the importance of antipodal invariance in our search space, proposing the use of the CS for exhaustive search whenever time is not a constrain (i.e. during training).

We propose a novel transform which we denote as HHC which boosts the antipodal invariance in the Euclidean space, and that we embed in the Spherical Hashing algorithm of Heo et al. [31], thus obtaining an Antipodally Invariant Spherical Hashing.

In order to further improve performance, we introduce a novel feature transform that performs better than previous gradient features thanks to a better coarse approximation of the upscaled gradients. The regressors obtained with an antipodally invariant metric show a neat gain in PSNR over those obtained with Euclidean distance and, furthermore, our antipodal SpH is very well adapted to the NN search, as the loss in quality when compared to exhaustive search is minimal.

Chapter 7

Naive Bayes SR Forest

7.1 Introduction

In Chapter 5 we introduced the usage of Spherical Hashing as a mean to alleviate the computational cost of the nearest neighbor search involved in piecewise linear regression SR methods. In Chapter 6 we discussed the importance of the antipodal symmetries and how to embed the data points so that the resulting distances are antipodally invariant in the Euclidean space. In this chapter we present a novel SR algorithm that tackles the same objectives (e.g. sublinear search and antipodal invariance) by means of a tree ensemble with antipodally invariant bimodal splits. Additionally, we revisit the NBNN selection procedure presented in Chapter 4 and adapt it to perform a per-patch selection of the best-suited tree within the forest, which leads to improvements in both speed and quality performance.

7.2 Hierarchical manifold learning

Tree structures are naturally hierarchical: There are $\mathcal{O}(n)$ split functions in a tree of n leafs, which is a fast growth when compared to the characteristic $\mathcal{O}(\log_2 n)$ of most hashing schemes. In tree structures, each node has a different split function and is normally trained with the data arriving to that specific node.The benefits of such hierarchical structure are specially valuable when used not only for fast nearest neighbor search during inference, but also in order to perform the unsupervised clustering task that provide the linearization anchor points and the neighborhoods to train the regression functions.

Our work is related to other hierarchical manifold learning approaches as the in-place SR of Yang et al. [93], as both methods use hierarchical structures for unsupervised clustering and fast inference, attaching a locally linear mapping in each leaf that conforms the piecewise linearization of the manifold mapping. As in DLT and HHC SR methods, the learning objective of the regression functions is the residual $(x - c)$, i.e. the difference between the HR image x and a coarse approximation c obtained through IBP , Equation (6.7).

The In-Place example regression of [93] fuses a unimodal tree that split the data

based on the thresholding of a certain data projection. This is the mechanism underlying the PCA tree [48], its random projection approximation [26] and also the faster k-D tree. In the latter, a set of features is precomputed for all data and the spliting is based on the thresholding of the most sensitive feature for each node, whereas the PCA tree and its approximation provide an adaptive computation of relevant features during the root-to-leaf tree traversal.

These unimodal trees tend to generate unbalanced partitions: the set of data lying out of the inclusion partition (projection above threshold) is much more heterogeneus than the one lying inside (below threshold), as we shown in Figure 7.1. In order to better represent the partition our data we propose a bimodal tree partitioning (Figure 7.1, bottom).

7.3 Antipodality and bimodal trees

As seen in Section 6.2, antipodally invariant metrics are of great importance within regression-based SR. Instead of using a embedding transform as in HHC, we design a tree split function that is able to group the two most relevant clusters of antipodal patches at each node.

To properly deal with antipodality when spliting our data, we use the CS described in Equation (6.2). During training, in each node we obtain two cluster centroids via the spherical k-means (i.e. adaptation of k-means that is able to handle antipodal data) [32, 15], namely $\{\mu_1,\mu_2\}$. Our split criterion for a node is defined as:

$$\mu^* = \underset{\mu_i=\{\mu_1,\mu_2\}}{\arg\max} \, |\mu_\mathbf{i} \cdot \mathbf{c}| \,, \tag{7.1}$$

where the centroid μ^* which has higher cosine similarity is the binary branch selected for a given datapoint $\mathbf{c}$.

7.4 Naive Bayes Super-Resolution Forest

Regression forest have been widely applied to several low-level and computer vision problems [10, 59, 39, 9, 8], even though it has not been applied to SR until fairly recently [68]. Our approach to SR Forest is inspired by the work of Bernard et al. [5], in which there is a tree selection strategy.

Combining all the trees in the ensemble might not yield the best possible performance. On the one hand, if the output of all the trees is combined the execution grows linearly with the number of trees (i.e. each tree requires computing a linear regression per input patch). On the other hand, combining all the trees in the ensemble might not always improve the results as some of the trees might deteriorate the performance [5].

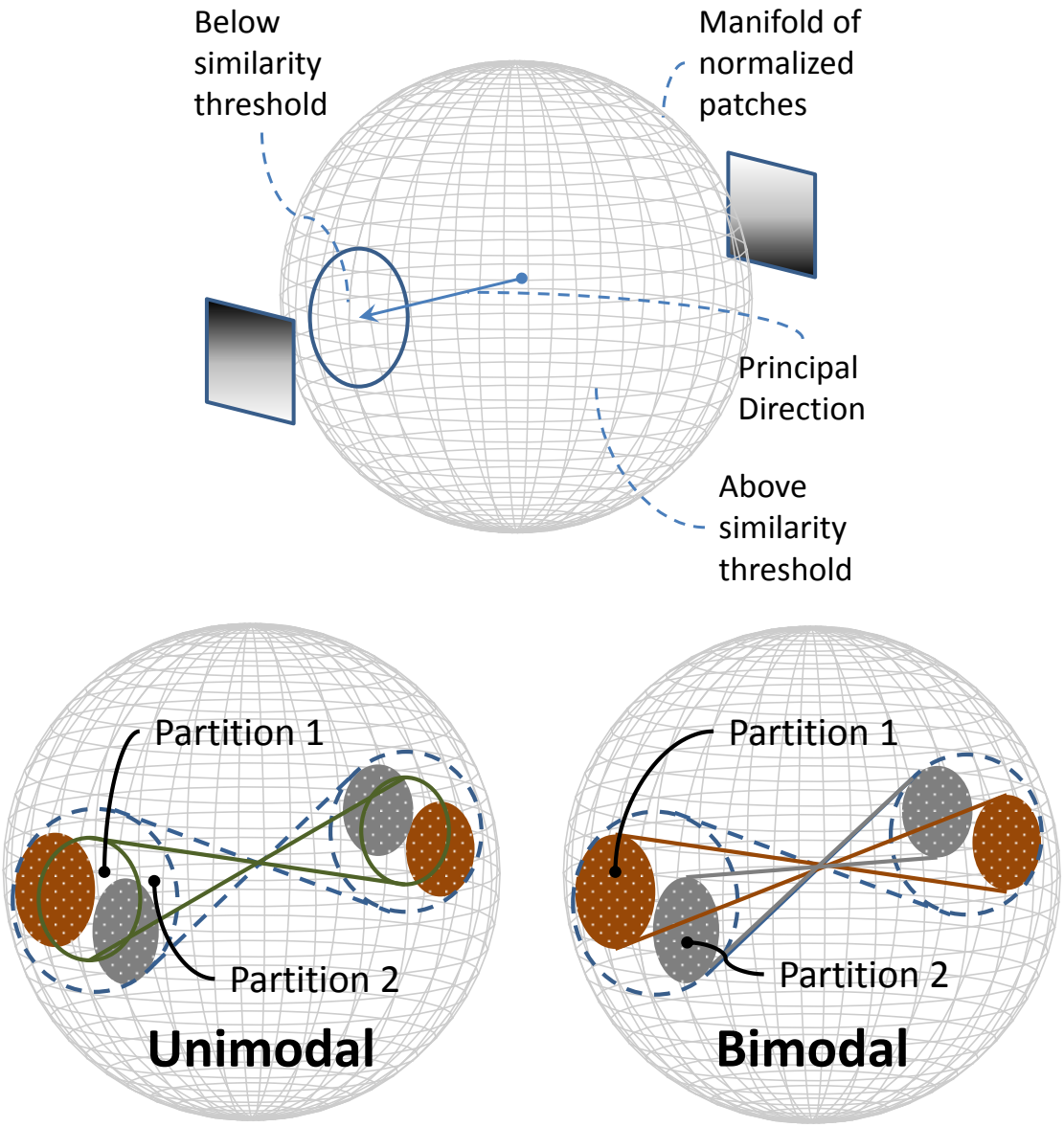


Figure 7.1: Top: Unimodal partitions on a normalized-feature space. The spherical cap (delimited by the solid line) is the fraction of the manifold that can be described with a single principal direction. Bottom: Antipodally invariant partitioning with unimodal and bimodal split functions. Bimodal partitioning is able to better separate the two data clusters (represented by red and gray doted caps).

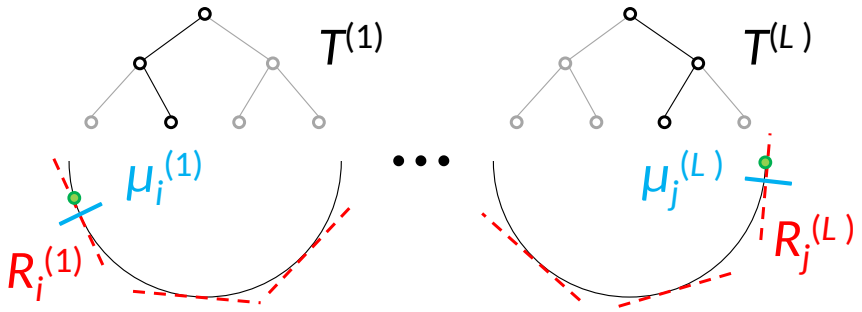


Figure 7.2: In NBSRF we select the tree T^k that provides the best local linearization of the mapping function for a given datum.

For computational complexity reasons we aim at selecting just one tree within our forest (i.e. only one regression is computed). We define the Naive Bayes Super-Resolution Forest (NBSRF) as a tree ensemble where the tree selection is based on the Naive Bayes assumption [7]. In Figure 7.2 we illustrate the advantages of using an ensemble of trees in this manner. If we are able to quantify the selectivity of each tree $T^{(k)}, 1 \leq k \leq N_{tr}$ with respect to an input datum $\mathbf{c}$ (green dot in the figure), we can perform a much more accurate regression than that attainable by considering a single tree. A straightforward solution to this problem consist in choosing the tree for which CS is maximum. Nonetheless, this criterion would discard the valuable information stored in the space partitions that leads to the leaf node in each tree, and it does provide suboptimal performance (see Table 7.1).

7.5 Von Mises-Fisher distribution

We model the data distribution at each node as a Von Mises-Fisher distribution [22] in order to evaluate the selectivity of a tree with respect to the datapoint. Von Mises-Fisher models a dispersion distribution over a unit hypershpere:

$$f(c,\mu,\vartheta) = \mathcal{C}(\vartheta)\exp(\vartheta(\mu \cdot \mathbf{c})), \tag{7.2}$$

where μ is a mean direction, ϑ is a concentration parameter that determines the dispersion from the central mode and $\mathcal{C}(\vartheta)$ is a normalizing constant. If we include CS metric to account antipodal data we obtain:

$$f(c,\mu,\vartheta) = \mathcal{C}'(\vartheta)\exp(\vartheta\,|\mu\mathbf{c}|), \tag{7.3}$$

where $\mathcal{C}'(\vartheta)$ normalizes the modified distribution. For each node in the tree, the data distribution is composed by a mixture of two antipodal Von Mises-Fisher distributions. We assume that both components in the mixture have the same concentration ϑ.

7.6 Local Naive Bayes tree selection

We use the same Local Naive Bayes framework [47] as in our adaptive region selection scheme (Chapter 4). We find the best-fitted regressor $R_i^{(k^*)}$ from tree $T^{(k^*)}$ as follows:

$$R_i^{(k*)} = \arg\max_{R_i^{(k)}} p(R_i^{(k)}|c) = \arg\max_{R_i^{(k)}} \log p(c|R_i^{(k)}). \tag{7.4}$$

The number of node responses or features $f_l^{(k)}$ equals the tree depth T_d and are computed from patch c in each root-to-leaf tree traversal. The problem with this formulation is that it requires computing the likelihoods for all possible paths across the tree. Fortunately, the alternative formulation by [47] allows us to avoid such exhaustive MAP. The effect of each node response to a patch c can be expressed as a log-odds update. This is extremely useful for trees, since it allows us to restrict updates to only those nodes for which the descriptor gives significant evidence (i.e. the visited nodes along the root-to-leaf traversal).

Let $R_i^{(k)}$ be some linear mapping and $\bar{R}_i^{(k)}$ the set of all other linear mappings. The odds $\mathcal{O}$ for the mapping $R_i^{(k)}$ with uniform priors is given by:

$$\mathcal{O}(R_i^{(k)}) = \frac{p(R_i^{(k)}|c)}{p(\bar{R}_i^{(k)}|c)} = \prod_{l=1}^{T_d} \frac{p(f_l^{(k)}|R_i^{(k)})}{p(f_l^{(k)}|\bar{R}_i^{(k)})}. \tag{7.5}$$

The final classification rule expressed as log-odds increments reads:

$$R_i^{(k*)} = \arg\max_{R_i^{(k)}} \sum_{l=1}^{T_d} \log \frac{p(f_l^{(k)}|R_i^{(k)})}{p(f_l^{(k)}|\bar{R}_i^{(k)})}, 1 \leq k \leq N_{tr}. \tag{7.6}$$

For each input patch c, we need to do N_{tr} root-to-leaf traversals, then apply the rule in Equation (7.6) to find the best-fitted tree within the ensemble, and finally apply the related regressor $R_i^{(k^*)}$ to it.

7.7 Validation

In Table 7.1 we show in more detail the effect of the contributions of the paper on $\times 2$ scaling on Set14 for increasing numbers of trees with a $T_d = 11$ (2048 leaves). The first columns of [68] and NBSRF essentially show the improvement of our bimodal clustering strategy. In the *bicubic* experiment we use NBSRF with bicubic interpolation instead of IBP to show the relative impact of the latter. With the *random* experiment we show that the Local Naive Bayes criterion of NBSRF is clearly better than a random tree choice (note that the latter is practically equivalent to having 1 tree), and with the *leaf* experiment we show that it is not sufficient to just observe the similarity between data and leaf modes (note that the performance is

Table 7.1: Different configurations of tree selection compared to our local Naive Bayes approach and the Super Resolution Forest of Schulter et al. [68].

	1 tree	2 trees	4 trees	8 trees	16 trees
SRF [68]	32.11	32.16	32.21	32.21	32.22
bicubic	32.28	32.32	32.34	32.35	32.35
random	32.38	32.39	32.40	32.39	32.39
leaf	32.38	32.40	32.40	32.41	32.40
average	32.38	32.43	32.45	32.46	32.45
NBSRF	32.38	32.42	32.44	32.45	32.46

only slightly better than that achieved with a single tree). In other words, we need to exploit all the root-to-leaf computed features, as in NBSRF, to choose the optimal tree. Finally, the *average* experiment shows that carrying out all the regressions and averaging (classical random forest) provides in practice the same accuracy, yet it is N_{tr} times costlier. The last column of this comparison shows that tree selection can eventually outperform averaging.

7.8 Summary and discussion

We present a novel method for example-based SR which we name NBSRF, aiming to high performance in both quality and runtime. NBSRF is essentially a hierarchical manifold learning approach that uses trees with bimodal split functions, where antipodal patches are effectively clustered together and both children subnodes have comparable homogeneity, thus leading to an overall better space sampling. We propose to use tree ensembles in a fast an efficient way by selecting the optimal regression tree based on a Local Naive Bayes criterion. By only selecting the optimal tree for every input patch instead of averaging over the output of all trees we to obtain the benefits of random regression forest with almost no extra computational complexity.

Chapter 8

Dihedral Symmetry Collapse

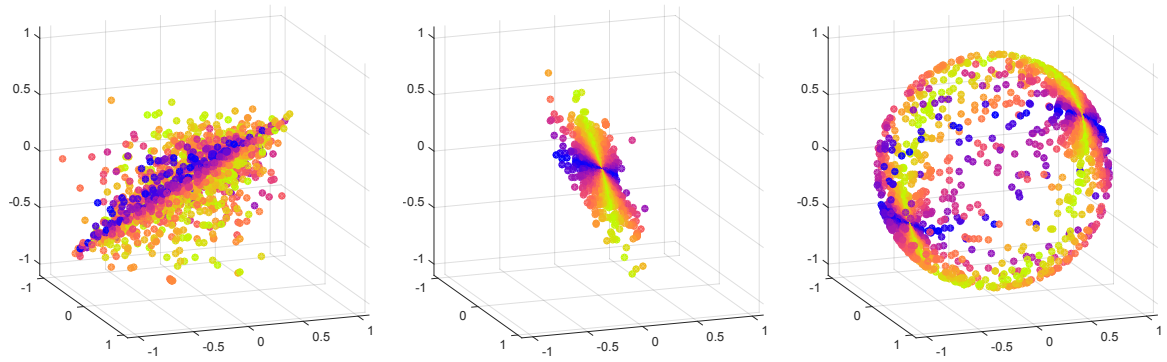
8.1 Introduction

Finding meaningful examples for SR is crucial both for internal learning (where the search space is limited by the image) and external learning. In this direction, Timofte et al. [76] proposed to generate new training data from different multi-scale images, Zhu et al. [100] proposed to deform patches based on optical flow and, more recently, Huang et al. [34] incorporate 3D scene geometry for cross-scale self-similarity using a modified PatchMatch [4].

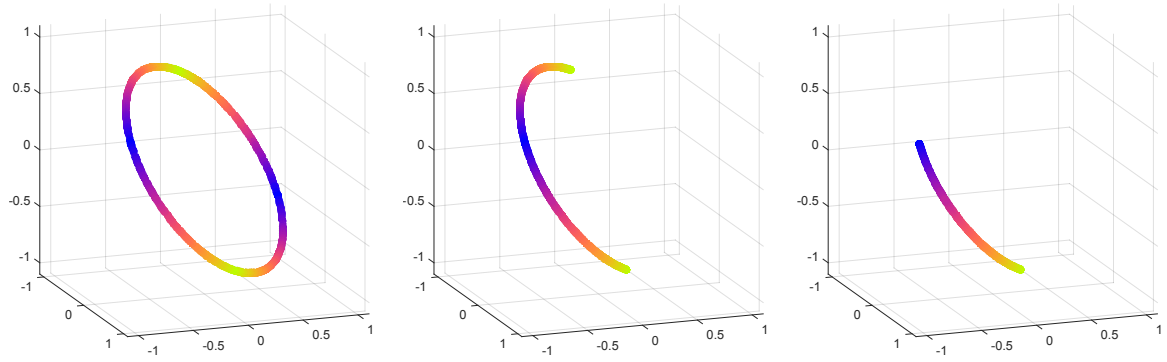
Another approach to improve the NN search consists in reducing variability of the manifold through the knowledge of its redundancy. In the early work of Freeman et al. [25], the concept of improving the NN search through the *collapse* of the manifold's variability was already addressed. In their learning process, to predict the highest frequency band they only consider the mid-frequency band and discard the rest of low-frequencies, thus *collapsing the training data for all possible low-frequency values into one value*. Similarly in concept, when subtracting the mean to a patch, all possible means are mapped to a single 0-mean patch. The benefit of removing the undesired variability of the manifold versus generating more data is obvious as the first one obtains the same advantages while not increasing the number of search candidates. In this chapter we further deepen the knowledge of the natural image patch manifold, analyzing the redundancy present within the manifold due to the dihedral group of transforms (i.e. rotation, vertical and horizontal reflections), which are invariant across scales and easily invertible (i.e. a lossless $f^{-1}(x)$ exists).

8.2 Reducing the manifold span

In this section we first overview two basic patch pre-processing steps (mean subtraction and normalization) and their effects within the manifold, followed by a geometric transformation model that can reduce the manifold span (extended in



(a) Raw patches (b) Mean subtraction (c) Normalization



(d) Mean sub. + norm. (e) Antipodal sym. (f) Reflection sym.

Figure 8.1: Reduction of the manifold's span and complexity by the procedures introduced in Section 8.2. The manifold is composed of three dimensional (i.e. 3×1) patches in the range of $[-1,1]$ extracted from images.

Section 8.3) and its analysis in the Discrete Cosine Transform (DCT) space. An overview of the presented transformation is shown in Figure 8.1.

8.2.1 Mean subtraction and normalization

Mean subtraction is an inexpensive process widely adopted in SR applications, as it is specially beneficial since the mean presents no variations across scales. Bevilaqua et al. [6] concluded in their feature analysis that the centered luminance patches are the best suited for their non-negative neighbor embedding SR. Within the manifold structure, mean subtraction collapses all the possible patches to lie on the hyperplane $\mathbf{1}^\intercal\mathbf{x} = 0$, as shown in Figure 8.1b.

Patch normalization is also a simple yet effective process very present in low-level vision, often interpreted as an illumination normalization. Patch normalization removes the undesired variability derived from scalar multiplication: All positive scalar variations are represented by a single unitary vector (i.e. a certain patch structure). In terms of manifold transformation, normalization enforces the patches to lie in the unitary hypersphere, as we show in Figure 8.1c. The combination of both mean subtraction and normalization limits the span of the manifold to the intersection of the mean hyperplane and the unitary hypersphere, a ring in the 3-dimensional example of Figure 8.1d.

8.2.2 Antipodality

Antipodal points (i.e. points that are diametrically opposed in the unitary sphere: $\mathbf{x}_A = -\mathbf{x}$) cannot be properly collapsed by patch normalization as norms are strictly positive, so any two normalized antipodal points are located at the furthest away Euclidean distance (the diameter of the hypersphere) while actually the structure of the patch is exactly the same (see Figure 8.2). In our previous work we already introduced antipodal invariance for SR (Chapter 6). It is possible to collapse antipodal variability together with dihedral transformations as described in Section 8.3 and illustrated in Figure 8.1e-8.1f.

8.2.3 Transformation models

Within the space of patches, numerous 2D geometric transformations have been proposed in order to model physical displacements in the 3D world, improve invariance to those transforms (e.g. rotation for object detection) or expand the search space both in testing and training. A general model for such transformations is the projective transformation model, also referred as *homography* or *collinearity*.

The projective transformation properly describes the possible transformations of a pinhole camera when moving to an arbitrary viewpoint. Homographies are widely used in several applications involving multiple cameras or camera motion [21, 29],

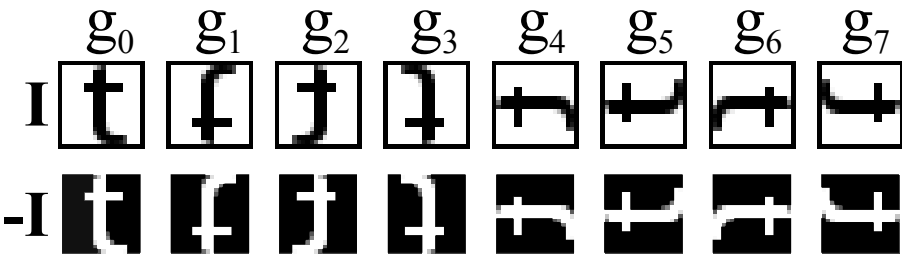


Figure 8.2: D_4 dihedral transforms applied to a 20x20 patch and its corresponding antipodal versions denoted with $-\mathbf{I}$.

and they have been also used recently in SR [35] in order to increase the number of relevant patches in the NN search.

Homographies show two main drawbacks when applied to SR. Firstly, as small patches present a very scarcely sampled grid, transforming its geometry requires interpolating values, which leads to a high-frequency loss. Secondly, the homography transform has 8 degrees of freedom, therefore being computationally expensive to explore and estimate (e.g. Huang et al. [35] use an affine transform enriched with some perspective deformation limited to a discrete set of detected planes).

We propose the usage of the dihedral group D_4 (for polygons of 4 sides, e.g. patches) [90], which is a subset of affine transformations that only includes rotations and reflections. This finite group $G = \{g_j\}_{j=0}^{7}$ contains 8 structure-preserving transforms which just re-distribute the elements within a patch and therefore do not require any interpolation. We can obtain the set of 8 dihedral transforms G via a combination of the following matrices in the 2D space:

$$g_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, g_y = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, g_\intercal = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{8.1}$$

where g_x and g_y denote the reflections along the x and y axis respectively, and $g_\intercal$ denotes the transpose operation. All the transforms forming the dihedral group are linear and scale invariant, and a straightforward inverse function exists. In Figure 8.2 we show the behavior of the dihedral group of transforms and how they affect a given patch.

8.2.4 Dihedral group in the DCT space

In this section we analyze the effect of the dihedral group G in the domain of the DCT, as there are some useful properties that lay the groundwork for our proposed method. The DCT b of a patch x of size $M \times N$ reads:

$$b(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) \cos \frac{\pi(m+\frac{1}{2})k}{M} \cos \frac{\pi(n+\frac{1}{2})l}{N}. \tag{8.2}$$

As the DCT is linear, applying the transpose operator (i.e. $g_\intercal$ in the 2D space) results in a transpose in the transformed space, i.e. $b^\intercal = f_{DCT}(x^\intercal)$. As for the reflection operators (i.e. g_x and g_y in the 2D space), they result in a change of sign in some of its components:

$$\begin{aligned} b_{g_x}(k,l) &= b(k,l) \cdot (-1)^l \\ b_{g_y}(k,l) &= b(k,l) \cdot (-1)^k \end{aligned} . \tag{8.3}$$

The behavior of the proposed dihedral transforms in the DCT space is therefore reduced to transpositions and sign changes in a defined set of coefficients. Figure 8.4 left shows which components of the DCT are expected to change whenever there is a reflection or transposing operator. This simple and predictable behavior in the DCT space facilitates the observation of mirror symmetries.

8.3 Manifold symmetries

The transform group G presented in Section 8.2.3 defines 8 points in the $M \times N$-dimensional manifold of natural patches for a given patch primitive x (see Figure 8.4 right). This is a dihedral symmetric shape within the manifold surface, since a symmetric structure is defined if there exists a non-trivial group of action that defines an isomorphism. Our goal is to exploit the symmetries defined by G together with antipodality in order to efficiently collapse redundant variability of our manifold span.

Our proposed Symmetry-Collapsing Transform (SCT) builds on the work of Zabrodsky et al. [96], where they proposed a continuous Symmetry Distance (SD) which measures how symmetric a given structure is. This metric δ is defined in the shape space Ω, where each shape is represented by a sequence of r points $\{P_i\}_{i=0}^{r-1}$. The metric reads:

$$\delta(P,Q) = \frac{1}{r} \sum_{i=0}^{r-1} \|P_i - Q_i\|^2 , \tag{8.4}$$

which is an averaged point to point Euclidean distance. In order to achieve invariance to symmetry, a Symmetry Transform (ST) of a shape P is defined as the symmetric shape closest to P in terms of Equation (8.4), and thus SD is defined as $SD = \delta(P, ST(P))$. The metric is therefore the point to point Euclidean distance of a given shape to its closest symmetric shape.

Zabrodsky et al. [96] present different ST depending on the type of symmetry to be accounted (e.g. rotational, mirror-symmetry). For the specific case of the mirror-ST, with a known mirror symmetry axis, the procedure for every pair of points $\{P_0, P_1\}$ is:

Fold by reflecting the point across the mirror symmetry axis obtaining $\left\{\mathring{P}_0, \mathring{P}_1\right\}$ (i.e. $P_0 \equiv \mathring{P}_0$).

Average both points to obtain a new average point A_0.

Unfold the average point A_0 in order to obtain A_1.

We show an overview of the original mirror-ST in Figure 8.3 (steps 1 and 2a). In the original algorithm, the ST aims to obtain a regular polygon which can be thereafter compared to the input shape in order to estimate its point to point distance. Our goal is to obtain a transform that reduces variability while respecting the SD.

To achieve this reduction, we present a modified ST, which we denote as SCT, that moves all the possible symmetric points to a reference side of the mirror axis, thus reducing redundant variability. For that purpose, assuming a single mirror axis, all the points are fold into the reference side where P_0 lies, and the element of the applied symmetry group (i.e. g_j) is saved. This is similar to a mean subtraction , where all possible different means of a given patch are collapsed to a single 0-mean patch and the mean is saved in order to differentiate among them. We show an overview of our proposed SCT in Figure 8.3 (steps 1 and 2b), where we highlight that the resulting distances are conserved with respect to the original algorithm. Although folding the points back to their original position is not necessary for the distance calculation in our SCT, we can do it at any point as the inverse SCT.

The initial ST and SD extend to any finite point-symmetry group G in any dimension, where the folding and unfolding are performed by applying the group elements [96]. However, when extending to more than 3D, finding the symmetry axes that minimize SD is non-trivial.

In order to (a) keep the transform under a reasonable complexity, (b) easily and analytically find the mirror axes of G and (c) benefit from behavior of G in the DCT domain, we propose a representation based on the first vertical and horizontal harmonics $b(1,0)$ and $b(0,1)$. Each of these coefficients is affected only by one reflection and the transpose is plainly mapped to a coefficient switch. Semantically, $b(1,0)$ and $b(0,1)$ are the coefficients statistically containing more energy that represent the response to vertical and horizontal variations, resembling the original vertical and horizontal 2D space of Zabrodsky et al. [96]. The three resulting mirror planes are straightforwardly obtained as $b(1,0) = 0$, $b(0,1) = 0$ and $|b(1,0)| - |b(0,1)| = 0$, as shown in Figure 8.4 right. At this stage, there is still ambiguity within this projected space as an antipodal point can be confused by a patch affected by vertical and horizontal reflections (as both vertical and horizontal coefficients have a sign change). In order to disambiguate, we include another dimension and a fourth mirror plane in $b(3,3) = 0$ which is not affected by transpose, nor vertical or horizontal reflection (as it is a DCT base with inner dihedral symmetry). This fourth axis, which we fold in the first place, represents the negative unitary matrix $-\mathbf{I}$ (i.e. sign change) to be applied both patch-wise and within the DCT domain before collapsing the rest of symmetries.

The final proposed transform $\mathring{c} = \kappa(c,\varphi(c))$ produces collapsed patches (denoted by the ring accent) using the four defined axes, where $g_j = \varphi(c)$ retrieves the element

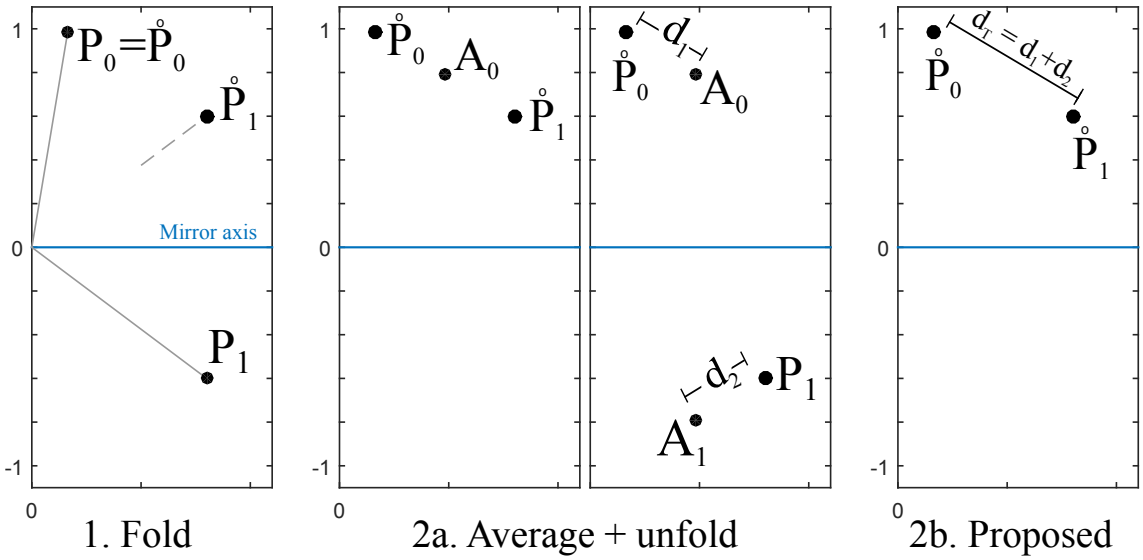


Figure 8.3: Mirror-Symmetry Transform of a single pair of points as proposed by Zabrodsky et al. [96] (1 and 2a) and our proposed SCT (1 and 2b).

within the group G together with the disambiguation of the sign (i.e. $-\mathbf{I}$ when $b(3,3) < 0$). The inverse $c = \kappa^{-1}(\mathring{c},\varphi(c))$ applies the same elements of the symmetry group that were used in the collapse in a reverse order, restoring the patch to its original appearance.

8.4 Application to SR

In this section we propose a novel SR algorithm that makes use of our proposed $\mathring{c} = \kappa(c,\varphi(c))$, which we name *Patch Symmetry Collapse (PSyCo)*. We denote 0-mean patches with the line accent (e.g. $\overline{\mathbf{c}}$). The main idea is to train our regression ensemble (both k anchor points in $\mathbf{D_l}$ and the associated regressors $\{R_i\}$) with the ground truth and coarse collapsed patches $\{\mathring{\mathbf{x}},\mathring{\mathbf{c}}\}$ so that during training time the system is optimized for the reduced span of the manifold which is to be used. We obtain our coarsely approximated images C with IBP. The kSVD input is a matrix of 0-mean patches without symmetric redundancy which have been stacked as columns, denoted by $\mathring{\overline{\mathbf{C}}}$. After that, a NN search with the angular similarity $\left|\mathring{\overline{\mathbf{d}}}_i^{\intercal}\mathring{\overline{\mathbf{c}}}\right|$ is performed for each atom $\mathbf{d}_i$ in $\mathbf{D}_l$ to construct each neighborhood $\mathbf{C}_i$ as a fixed-size subset of the whole training data $\mathbf{C}$. Once the anchor points and neighborhoods have been defined, each regressor R_i is trained with the following closed-form expression:

$$R_i = (1+\lambda)(\mathring{\mathbf{X}}_i - \mathring{\mathbf{C}}_i)\mathring{\overline{\mathbf{C}}}_i^{\intercal}(\mathring{\overline{\mathbf{C}}}_i\,\mathring{\overline{\mathbf{C}}}_i^{\intercal} + \lambda\mathbf{I})^{-1}. \tag{8.5}$$

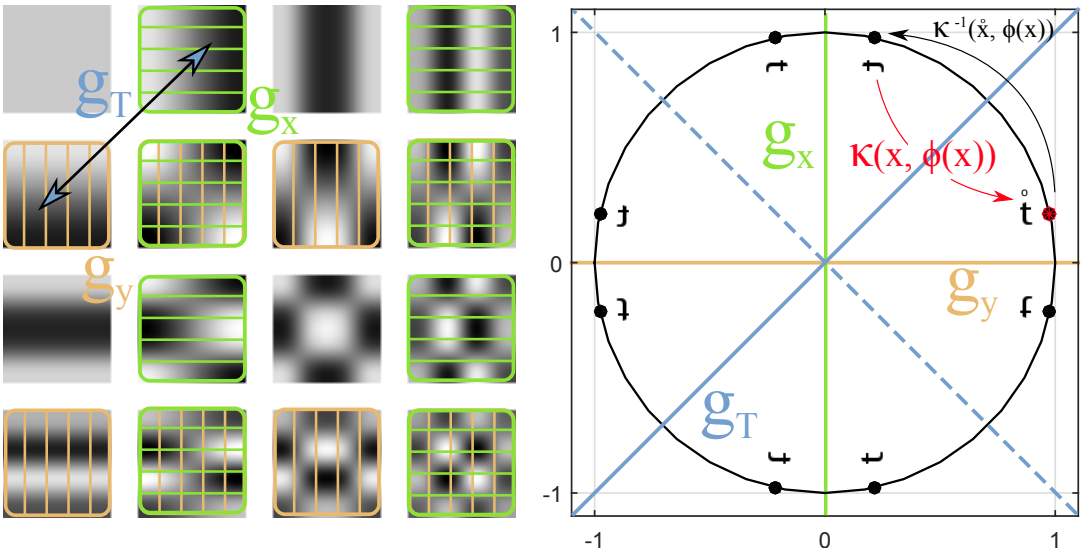


Figure 8.4: **Left**: Coefficients of a DCT that are affected by g_x, g_y (resulting in a sign change, Equation 8.3) and g_T (resulting in a transpose of coefficients). **Right**: Overview of our $\kappa(x,\varphi(x)$ with real patches, highlighting the symmetry axes associated to each operator.

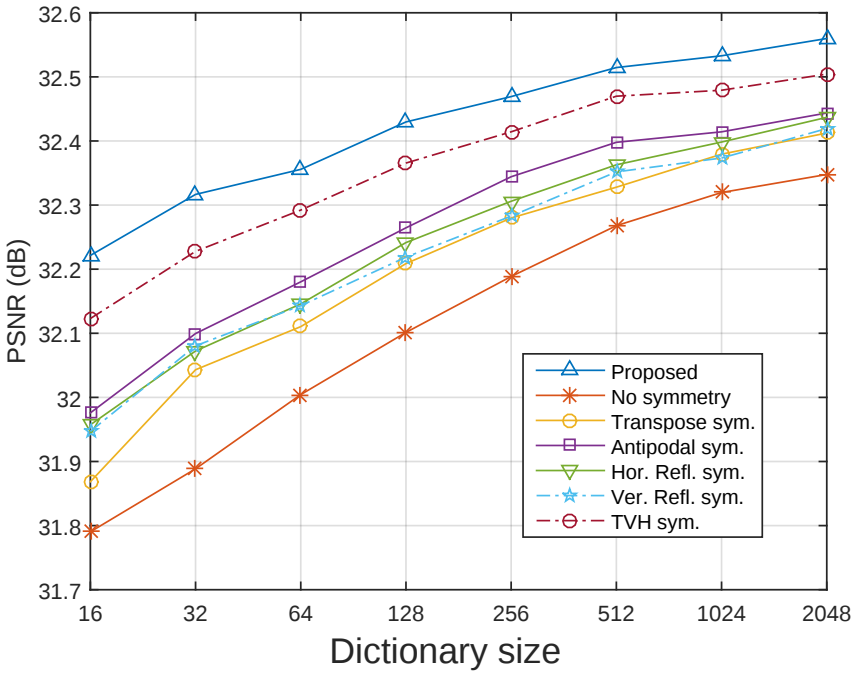


Figure 8.5: PSNR performance for different mirror symmetries accounted.

During inference time, the NN search and regression is performed with $\left\{\mathring{\overline{\mathbf{d}}},\mathring{\overline{\mathbf{c}}}\right\}$ and after regression the symmetric transformation needs to be reverted so that the patches recover their original orientation. The regression stage reads:

$$\tilde{\mathbf{x}} = \mathbf{c} + \kappa^{-1}(R^* \, \mathring{\overline{\mathbf{c}}}, \varphi(\overline{\mathbf{c}})), \tag{8.6}$$

and the final image $\tilde{X}$ is obtained by an overlapping reconstruction strategy, as it is common in SR [74, 92, 68].

8.5 Validation

In this section we validate the contributions of our proposed transform, assessing the impact of collapsing each of the axes separately, and also the combination of those exclusively corresponding to the dihedral group G and the impact of the complete system, which also tackles antipodal symmetries. Figure 8.5a shows *PSyCo* with several mirror-axes configurations and dictionary sizes. First, we would like to asses the benefits of our symmetric transform when compared to untransformed patches. The quality is around 0.4 dB higher for small dictionary sizes (e.g. 16, 32) and around 0.2 dB for 1024 atoms. We find remarkable the fact that our symmetry transform performs always slightly better than a $\times 16$ times larger dictionary without any symmetry accounted. This supports the idea that with our manifold collapse we can effectively cover the 16 different appearances of a given primitive patch without increasing the search space, plus an additional quality gain as the training of the regressors is better (i.e. due to more meaningful patches in the neighborhoods).

When it comes to assess the incidence of each type of transform separately, we find that all have similar impact, being the antipodal symmetry slightly better-performing than the reflection or the transpose. We also note that each symmetry axis is roughly comparing equally to a $\times 2-4$ times larger untransformed dictionary. The dihedral symmetries together surpass that of the antipodal, and we observe that its quality performance surpasses by a great margin that of the $\times 8$ larger dictionary without any symmetry.

8.6 Summary and discussion

In this chapter we present the last contribution of this thesis, which in a certain way presents a unifying framework for manifold learning with antipodal and dihedral invariance. We present a novel regression-based SR algorithm that benefits from an extended knowledge of the structure of both LR and HR patch manifolds. We propose a transform that collapses the 16 variations induced from the dihedral group of transforms (i.e. rotations, vertical and horizontal reflections) and antipodality (i.e. diametrically opposed points in the unitary sphere) into a single primitive. The

key idea of our transform is to study the different dihedral elements as a group of symmetries within the high-dimensional manifold. We obtain the respective set of mirror-symmetry axes by means of a frequency analysis of the dihedral elements, and we use them to collapse the redundant variability through a modified symmetry distance. The experimental validation of our algorithm shows the effectiveness of our approach, which obtains competitive quality with a dictionary of as little as 32 atoms (reducing other methods' dictionaries by at least a factor of 32) and further pushing the state-of-the-art with a 1024 atoms dictionary.

Chapter 9

Results

In this section we introduce the set-up used for the experimental validation of the dissertation, i.e. the metodolody, metrics and datasets used for our experiments. All the relevant methods described in the thesis are then separately evaluated and their parameters analyzed and discussed. A final section does a complete benchmarking of all the algorithms and compares the different metrics established for the experimental validation.

9.1 Methodology

Our experimental set-up aims at assessing the performance of image-upscaling algorithms. In order to be able to objectively evaluate the quality of the upscaled image, a reference ground truth image is required. Most of the well-known quality assessment metrics require pixel-wise error measurements, therefore the reference image and the output image should be correctly alligned, as even subpixel shifts create great disturbances in such measurements. In order to satisfy both requirements, we downscale the reference ground truth images by a given magnification factor S (i.e. 2, 3 and 4) with a bicubic kernel and then apply the SR algorithms to restore the image to the original resolution. This procedure represents the SR reconstruction constrain described in Section 2, Equation (2.1), where the degrading kernel H is the anti-aliasing bicubic filtering. We obtain our estimated HR image $\tilde{X}$ via SR, and we then compute several metrics related with quality assessment, and also the processing time and memory usage of the SR algorithm itself. We show an overview of the presented methodology in Figure 9.1.

9.2 Metrics

9.2.1 Peak Signal-to-Noise Ratio

PSNR describe the ratio between the maximum possible power value of the signal and the power of the noise that corrupts it. PSNR is defined in logarithmic scale

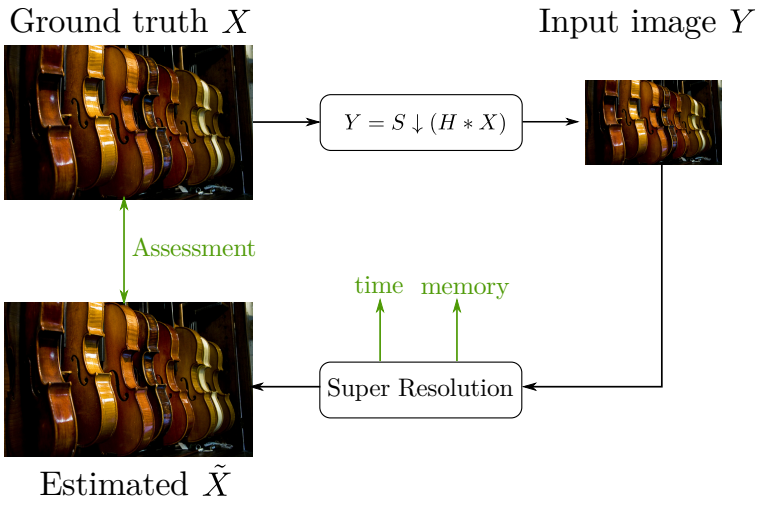


Figure 9.1: Overview of the experimental methodology. The reference ground truth image X is downscaled and filtered with bicubic kernel H. This new LR image Y is processed by the SR algorithm, from which we measure time and memory usage. The estimated HR image $\tilde{X}$ is compared with the reference image X to obtain objective error measurements.

and it is expressed in decibels (dB). The noise power is computed by estimating the MSE with respect to the ground truth signal:

$$\text{MSE} = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}(S_{ref}(i,j) - S_{noise}(i,j))^2, \tag{9.1}$$

where S_{ref} is the clean reference ground truth image and S_{noise} is the observed noisy image. The PSNR metric relates this mean quadratic error to the maximum value of the signal:

$$\text{PSNR} = 10\log\frac{\text{MAX}^2}{\text{MSE}}. \tag{9.2}$$

We normalize the dynamic range of our images to be in the range [0,1], so that MAX is consistently 1.

PSNR and MSE are probably the metrics most widely used in SR benchmarking and, more generally, in image and video quality assessment. The formulas have very intuitive physical meanings, and they are additionally simple and fast to compute. From a mathematical perspective, minimizing over MSE is very well understood and a common approach in many state-of-the-art algorithms. As of today, there is also not a clear standardized alternative to these quadratic fidelity metrics [89].

Despite its wide adoption, PSNR has been shown to correlate poorly with the quality perceived by human viewers [91]. This is caused by the fact that data metrics like PSNR overlook the human visual system and rather compare only byte

to byte distances without any further intelligence. Wang and Bovik [85] describe some of the implicit assumptions of signal fidelity metrics (e.g. MSE and PSNR) which are specially problematic for image quality assessment:

1. PSNR is independent from temporal and spatial pixels relationships. If both the reference and distorted pixels are re-arranged in the same way, the PSNR remains the same.

2. PSNR is independent from any relationship between the error and the original signal. For a given error signal, PSNR remains unchanged regardless of which original signal it is added to.

3. PSNR ignores the sign of the error signal.

4. In PSNR all the samples are equally important to the global error computation.

All these four assumptions do not hold in the context of evaluating visual quality perception [89, 85]. Many other quality assessment metrics have been proposed in order to better relate to the visual human system. The Structural Similarity (SSIM) index of Wang and Bovik [86] represent a remarkable effort in that direction.

In the benchmark analysis of Yang et al. [91] they compare and study different metrics for the evaluation of SR techniques. They include subjective evaluation and compute how each of the metrics involved correlate with it. The metric that mostly correlates with subjective perception is the Information Fidelity Criterion (IFC), and thus we include this together with the widely adopted SSIM and PSNR for our SR evaluation.

9.2.2 SSIM

The SSIM index is based on the assumption that the human visual system is highly adapted to extract structural information from the viewing field, and thus measuring structural change can approximate perceived image distortion [86]. For a reference image X and a corrupted image C from which we obtain patches x and c respectively, SSIM index breaks the quality measures in three different components: luminance, contrast and structure.

The luminance component is compared using the means of both patches as follows:

$$l(\mathbf{x}, \mathbf{c}) = \frac{2\mu_x\mu_c + C_1}{\mu_x^2 + \mu_c^2 + C_1}, \tag{9.3}$$

where $\mu_x = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$ denotes the local mean of the patch $\mathbf{x}$ and C_1 is a constant to avoid numerical instability.

The contrast comparison function takes a similar form, but it uses the standard deviation $\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \mu_x)^2\right)^{1/2}$ as an estimate of the signal constrast:

$$co(\mathbf{x}, \mathbf{c}) = \frac{2\sigma_x\sigma_c + C_2}{\sigma_x^2 + \sigma_c^2 + C_2}. \tag{9.4}$$

Equation (9.4) is less sensitive to contrast change $\triangle\sigma = \sigma_c - \sigma_x$ for the case where there is high base contrast, which tries to reproduce the masking feature of the human visual system [86].

The structure comparison function is based on the correlation after luminance and contrast normalization:

$$s(\mathbf{x}, \mathbf{c}) = \frac{\sigma_{xc} + C_3}{\sigma_x\sigma_c}, \tag{9.5}$$

where σ_{xc} is estimated as:

$$\sigma_{xc} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}_i - \mu_x)(\mathbf{c}_i - \mu_i). \tag{9.6}$$

Equation (9.5) corresponds to the cosine of the angle between the vectors $(x_i - \mu_x)$ and $(x_i - \mu_x)$.

The SSIM index between signals is a composition of the three of Equations (9.3), (9.4) and (9.5):

$$\text{SSIM}(x, c) = [l(\mathbf{x}, \mathbf{c})]^a \cdot [co(\mathbf{x}, \mathbf{c})]^b \cdot [s(\mathbf{x}, \mathbf{c})]^c, \tag{9.7}$$

where a, b and c weight the impact of each of the components, and which are equally set to 1 in the original work of Wang and Bovik [86]. A symmetric Gaussian weighting function is applied in order to avoid block-like artifacts in the SSIM index map of the complete image, and the mean of all the values of the image is computed in order to represent the overall image quality.

9.2.3 IFC

The IFC of Sheikh et al. [70] is a quality assessment algorithm based on natural scene statistics that are analysed from an information theory perspective, i.e. modelled as a transmitter, channel and receiver. Images and videos of the three dimensional visual environment form a certain subspace in the space of all possible signals, and several efforts have been made to develop sophisticated models that characterize those statistics [73]. A given degradation of an image can be analyzed as a disturbance in those statistics. Sheikh et al. [70] propose the usage of natural scene statistics combined with distortion models in order to quantify the statistical information shared between the degraded and reference image.

The source model that they propose in their work is based on Gaussian Scale Mixture (GSM) [82] in the wavelet domain. We start defining one subband of the wavelet decomposition of an image as a GSM random field , $\mathcal{F} = \{F_i : i \in I\}$, where

I denotes the set of spatial indices for the random field and $\mathcal{C}$ is a product of two stationary random fields that are independent of each other:

$$\mathcal{C} = \mathcal{S} \cdot \mathcal{U}, \tag{9.8}$$

where $\mathcal{S}$ is a random field of positive scalars and $\mathcal{U}$ is a Gaussian scalar random field with mean zero and variance σ_U^2.

The distortion model is also described in the wavelet domain. It is a simple signal attenuation and additive Gaussian noise model in each subband:

$$\mathcal{D} = \mathcal{G}\mathcal{F} + \mathcal{V} = \{g_i F_i + V_i : i \in I\}, \tag{9.9}$$

where $\mathcal{F}$ denotes the random field from a subband in the reference signal, $\mathcal{D}$ denotes the random field from the corresponding subband from the test (degraded) signal, $\mathcal{G}$ is a deterministic scalar attenuation field, and $\mathcal{V}$ is a stationary additive zero-mean Gaussian noise random field with variance σ_V^2.

Given a source model and a distortion model, the information fidelity criterion is the mutual information between the source and the distorted images. Let $F^N = (F_1,F_2,\ldots,F_N)$ denote N elements from $\mathcal{F}$. Let Let $D^N = (D_1,D_2,\ldots,D_N)$ denote the corresponding elements from $\mathcal{D}$. The mutual information between these is denoted as $I(F^N,D^N)$. The information fidelity criterion proposed in [70] is the conditional mutual information $I(F^N;D^N|S^N = s^N)$, where $S^N = (S^1,S^2,\ldots,S^N)$ are the corresponding N elements of $\mathcal{S}$, and for single wavelet subband reads:

$$I(F^N;D^N|S^N = s^N) = \frac{1}{2}\sum_{i=1}^{N} \log_2(1 + \frac{g_i^2 s_i^2 \sigma_U^2}{\sigma_V^2}). \tag{9.10}$$

The IFC is then obtained by summing over all subbands:

$$IFC = \sum_{k \in subbands} I(F^{N_k,k};D^{N_k,k}|S^{N_k,k} = s^{N_k,k}),$$

where $F^{N_k,k}$ denotes N_k coefficients from the random field $\mathcal{F}^k$ of the k-th subband.

The IFC metric measures fidelity, not distortion, and therefore it ranges from zero (no fidelity) to infinity (perfect fidelity).

9.2.4 Time

In order to evaluate computational complexity we measure the time that a given algorithm takes to upscale an input LR image to an output HR image. All the experiments were run on an Intel Xeon W3690 @ 3.47GHz equipped with 12GB of RAM memory.

We also compute the frame frequency or frame rate that the algorithm can provide, which is a metric proportional to the speed of the algorithm and thus helps for

visualization purposes in our radar plots. The framerate f_{frame}:

$$f_{frame} = \frac{1}{t_{frame}}, \tag{9.11}$$

where t_{frame} describes the amount of time required to upscale a given frame.

9.2.5 Model Size

Different approaches to SR learn different data structures from the training data, e.g. regressors, anchor points, dictionaries, deep networks weights. For each of those algorithms it is also possible to modify certain parameters so that more information is learnt and stored (e.g. the number of dictionary atoms or hidden network layers), but this comes not only at a computational cost but also at a higher memory usage cost.

Some algorithms use very reduced sets of data and still are competitive in terms of quality performance. Also, in this thesis there is great emphasis on being efficient in the learnt representations thanks to the exploitation of inherent symmetries of patch manifolds.

In order to assess the improvements achieved in this direction, specially within algorithms that share similar data structures (e.g. regression-based SR), we measure the amount of memory $\mathcal{M}_{SR}$ necessary to store and use each algorithm for its recommended configuration. Together with the quality metrics, this can help interpret how efficient the learnt data is.

In order to visualize this values within spider plots we define the Memory Compression Ratio ϕ_2 (i.e. for magnification factor $\times 2$), which consist on a ratio that relates the size of the current model to the model size of the ANR SR in [74], so that the memory usage is inversely proportional to the Memory Compression Ratio and thus better memory usage (i.e. less bytes) translates to higher compression values. We reference to ANR as it is a well-known algorithm with a medium sized set of learnt data (regressors and anchor points), which is also directly related both to dictionary- and regression-based SR.

$$\phi_2(\mathcal{M}_{SR}) = \frac{\mathcal{M}_{ANR}}{\mathcal{M}_{SR}}. \tag{9.12}$$

We define $\mathcal{M}_{ANR}$ as the size in bytes (B) of the ANR SR in [74]:

$$\mathcal{M}_{ANR} = ((p_s^2 f_{PCA})d_s + (f_{PCA} d_s))s_{float}, \tag{9.13}$$

where d_s denotes the number of atoms (1024), p_s denotes the HR patch dimensionality, f_{PCA} denotes the LR feature dimensionality compressed via PCA and s_{float} denotes the size of the representation of each element (i.e. we assume all the elements to be 4Bytes floats).

Figure 9.2: Images from datasets Set5, Set14 and kodak.

9.3 Datasets

We use `Set5`, `Set14` and `kodak` datasets for our testing sets. `Set5` and `Set14` are composed by two sets of 5 and 14 images respectively and have been the reference datasets to compare in the recent state of the art benchmarks. `Set5` has images from 65kpixels to 262kpixels. `Set14` has images that range from 76kpixels to 393kpixels. The `kodak` dataset has not been adopted as widely as `Set5` and `Set14`, however it is a good option as it offers a set of 24 clean, high-quality images (24 bits per pixel) of a fixed resolution of 393kpixels. We show some examples of the images in the test datasets in Figure 9.2.

9.4 Sparse SR

In this section we provide performance evaluation for the sparse SR of Yang et al. [95] and the follow-up work of Zeyde et al. [97] which has an emphasis on efficiency and computing time.

Configuration

The original sparse algorithm of Yang et al. [95] requires setting several parameters. We use the recommendations of the authors and replicate their experimental setup: The regularization parameter λ (see Eq. 2.6) is set to $\lambda = 0.1$ (if there is noise presence, λ should increase proportionally to the standard deviation of the noise), the patch size is set to 5 pixels extracted in a full overlap scheme, the number of atoms in the trained coupled dictionary d_s is set to $d_s = 1024$ and the maximum number of iterations for the IBP (see Eq.(2.8)) is set to 20.

As for the method of Zeyde et al. [97], the main parameter to set is the size of

Table 9.1: Performance of ×2, ×3 and ×4 magnification in terms of averaged PSNR (dB) and average execution time (s) on datasets Set5, Set14 and kodak.

		Bicubic		Yang et al. [95]		Zeyde et al. [97]	
	S	PSNR	time	PSNR	time	PSNR	time
`Set5`	2	33.661	0.001	36.015	156.513	35.805	**4.202**
	3	30.392	0.001	31.928	143.225	31.912	**1.801**
	4	28.421	0.001	29.579	140.359	**29.694**	**1.060**
`Set14`	2	30.232	0.002	**31.946**	312.872	31.812	**8.309**
	3	27.541	0.001	**28.673**	287.218	28.669	**3.682**
	4	26.000	0.001	26.805	279.020	**26.879**	**2.190**
`kodak`	2	30.845	0.002	**32.330**	564.734	32.206	**14.556**
	3	28.426	0.002	**29.245**	514.468	29.221	**6.418**
	4	27.223	0.002	27.793	489.638	**27.837**	**3.786**

the dictionary, which we also fix to $d_s = 1024$. Additionally, the patch size is set to 3 pixels in the LR space, which is then multiplied by the magnification factor when extracting patches in the coarse upscaled image (i.e. for ×2 the patch size in the HR space is 6 pixels). The level of sparsity (number of elements which are non-zero) is set to 3 in the OMP and k-SVD algorithms. This parameter set is re-used in all the following experiments if not stated otherwise.

Performance

In Table 9.1 and 9.2 we show the objective evaluation (in terms of PSNR, IFC, SSIM and time) of these two methods compared with the bicubic interpolation which acts as a baseline. The method of Yang et al. has consistently a better performance for a ×2 magnification factor for the three measured quality metrics. The difference becomes tighter for ×3, and as for ×4 Zeyde et al. is the best performer in terms of PSNR, with a very similar performance in terms of SSIM and IFC. The execution times of Yang et al. are two orders of magnitude higher than those of Zeyde et al., thus the efficient sparse decomposition via OMP presented by [97] is specially effective.

In Figure 9.3 we show close-ups for visual inspection. This subjective evaluation is in consonance with the objective results, as images obtained with Yang et al. SR present sharper edges for ×2 upscaling, but for ×3 and ×4 there are strong artifacts along the edges. Images obtained with Zeyde et al. are slightly less sharp but also present less artifacts and therefore are more pleasant to the eye.

Table 9.2: Performance of ×2, ×3 and ×4 magnification in terms of averaged IFC and average SSIM on datasets Set5, Set14 and kodak.

		Bicubic		Yang et al. [95]		Zeyde et al. [97]	
	S	IFC	SSIM	IFC	SSIM	IFC	SSIM
Set5	2	6.282	0.930	**9.626**	**0.952**	8.204	0.950
	3	3.616	0.869	**5.056**	**0.899**	4.550	0.897
	4	2.342	0.811	**3.142**	0.840	2.953	**0.843**
Set14	2	6.304	0.869	**9.142**	**0.903**	7.939	0.899
	3	3.535	0.774	**4.713**	**0.812**	4.297	0.808
	4	2.259	0.702	**2.927**	**0.736**	2.755	0.734
kodak	2	6.223	0.870	**8.644**	**0.904**	7.649	0.900
	3	3.410	0.779	**4.367**	**0.811**	4.049	0.807
	4	2.126	0.719	**2.657**	**0.745**	2.540	0.744

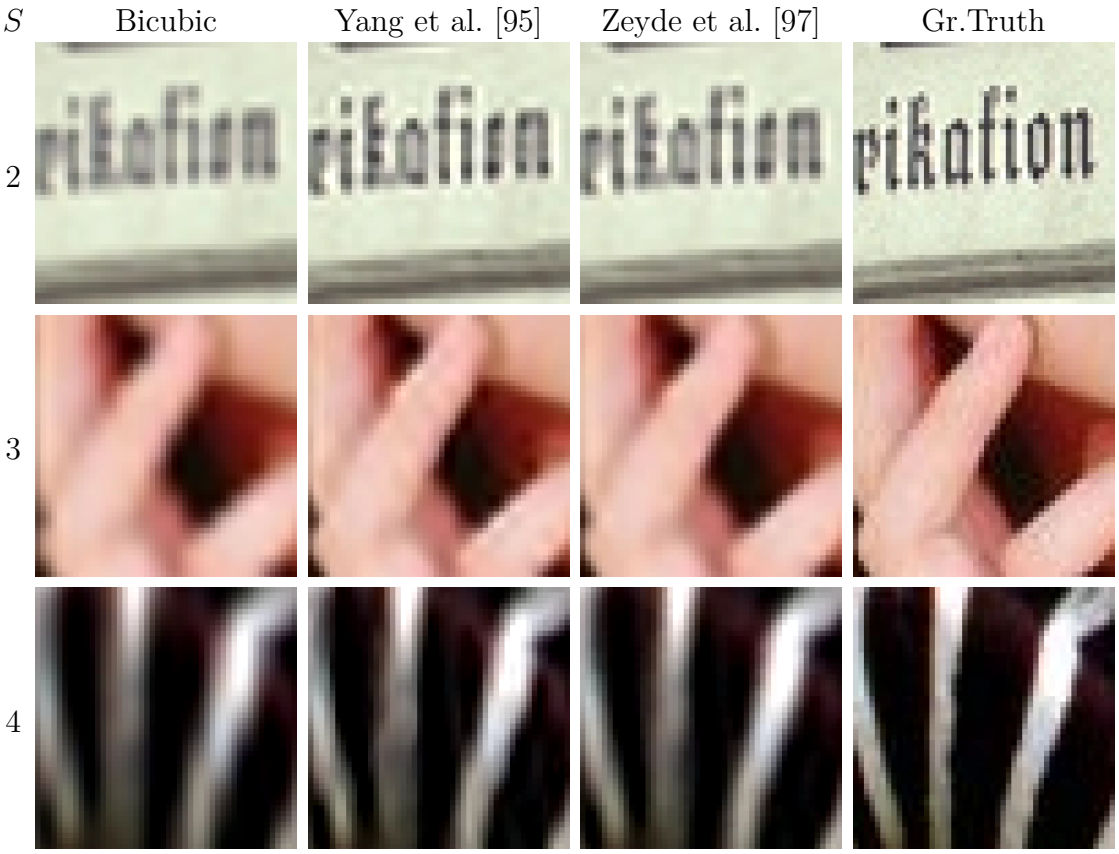


Figure 9.3: Close-ups of the results in Table 9.1 and 9.2 for visual qualitative assessment. Best-viewed zoomed in.

Table 9.3: Performance of ×2, ×3 and ×4 magnification in terms of averaged PSNR (dB) and average execution time (s) on datasets Set5, Set14 and kodak.

		Bicubic		Zeyde et al. [97]		GR		ANR	
	S	PSNR	time	PSNR	time	PSNR	time	PSNR	time
Set5	2	33.661	0.001	**35.805**	4.202	35.153	0.515	35.858	**0.783**
	3	30.392	0.001	31.912	1.801	31.423	0.282	**31.926**	**0.402**
	4	28.421	0.001	**29.694**	1.060	29.342	0.199	29.691	**0.273**
Set14	2	30.232	0.002	**31.812**	8.309	31.355	1.011	31.801	**1.519**
	3	27.541	0.001	**28.669**	3.682	28.305	0.568	28.647	**0.801**
	4	26.000	0.001	**26.879**	2.190	26.589	0.421	26.846	**0.564**
kodak	2	30.845	0.002	32.206	14.556	31.873	1.711	**32.245**	**2.573**
	3	28.426	0.002	**29.221**	6.418	28.987	0.968	29.213	**1.395**
	4	27.223	0.002	**27.837**	3.786	27.639	0.716	27.806	**0.975**

9.5 Anchored Neighborhood Regression

Configuration

We follow the author's parameter setting proposed in [74]. This comprises a dictionary size $d_s = 1024$ and a neighborhood size of 40 atoms. Other parameters such as patch size and maximal sparsity are set-up equally as in the previous work of Zeyde et al. [97].

Performance

In Table 9.3 and 9.4 we show the objective evaluation (in terms of PSNR, IFC, SSIM and time) of GR and ANR, and we compare them with the SR algorithm of Zeyde et al. [97]. The performance of ANR is similar to that of Zeyde et al., with differences upper bounded by 0.06dB. The IFC and SSIM indices of both algorithms are also comparable, only with marginal differences. Nonetheless, we would like to remark that ANR is notably faster, obtaining times about one order of magnitude faster (i.e. around ×4 times faster). The GR, which consist on a single regressor, yields lower PSNR and SSIM values, even though surprisingly is the best performer in terms of IFC. In terms of time is also slightly faster than ANR as it does not perform any nearest neighbor search.

In Figure 9.4 we show close-ups for visual inspection. GR produces heavy ringing artifacts, whereas both Zeyde et al. and ANR produce fairly similar images. All the images in the figure contain aliased frequencies, meaning that their schemes are not able to recognize and minimize its presence.

Table 9.4: Performance of ×2, ×3 and ×4 magnification in terms of averaged IFC and average SSIM on datasets Set5, Set14 and kodak.

		Bicubic		Zeyde et al. [97]		GR		ANR	
	S	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM
`Set5`	2	6.282	0.930	8.204	**0.950**	**8.663**	0.945	8.523	**0.950**
	3	3.616	0.869	4.550	**0.897**	4.649	0.885	**4.689**	**0.897**
	4	2.342	0.811	2.953	**0.843**	2.987	0.828	**3.030**	0.842
`Set14`	2	6.304	0.869	7.939	0.899	**8.433**	0.898	8.193	**0.901**
	3	3.535	0.774	4.297	0.808	**4.448**	0.803	4.415	**0.810**
	4	2.259	0.702	2.755	0.734	2.822	0.728	**2.829**	**0.735**
`kodak`	2	6.223	0.870	7.649	0.900	**8.168**	0.899	7.885	**0.902**
	3	3.410	0.779	4.049	0.807	**4.195**	0.804	4.145	**0.809**
	4	2.126	0.719	2.540	**0.744**	**2.599**	0.740	2.589	**0.744**

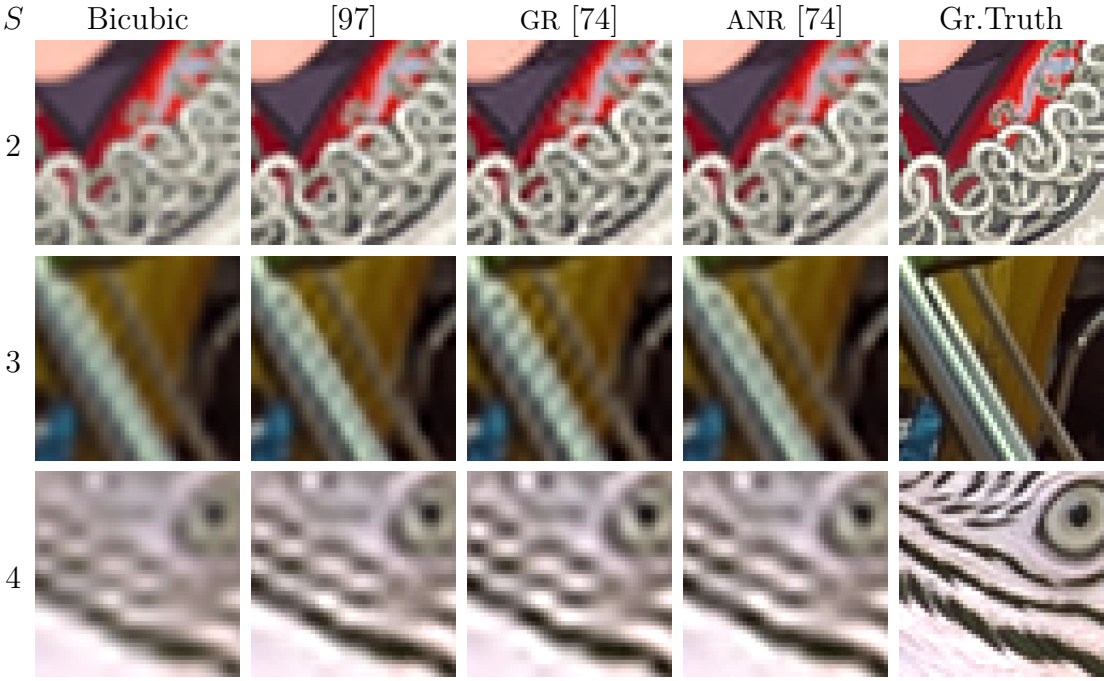


Figure 9.4: Close-ups of the results in Table 9.3 and 9.4 for visual qualitative assessment. Best-viewed zoomed in.

9.6 Adaptive dictionaries

Configuration

In our original publication [60] we used our adaptive dictionary training together with the scheme of [95]. Our local NBNN can be used together with any algorithm that builds a dictionary (e.g. the Naive Bayes Super-Resolution Forest introduced in Chapter 7). When used with sparsity, it requires building a dictionary per image or per sequence of images of similar content, thus efficient dictionary building have sensitive impact in the computational time. Consequently, we preferred for its evaluation within this dissertation the usage of the k-SVD for our sparse dictionary training and OMP for the sparse decomposition.

We set our dictionary size $d_s = 512$ in order to minimize the training time impact. We set the maximal sparsity to 3 elements. We compare the advantage of this adaptive dictionary with [97] also set to $d_s = 512$.

Performance

In Table 9.5 and 9.6 we show the objective evaluation (in terms of PSNR, IFC, SSIM and time) of our NBNN sparse SR compared with the efficient sparse SR of Zeyde et al. [97] and bicubic interpolation. Our proposed adaptive approach improves the performance over Zeyde et al. consistently for all the magnification factors and datasets, with increments up to 0.27dB. The IFC and SSIM indices are also consistently improved, with a remarkable gap in terms of IFC of 1.4 for the $\times 2$ factor. If we compare the performance of our adaptive 512 atoms dictionaries with Zeyde et al. [97] trained with 1024 atoms (see Table 9.1) we can see that even with half sized dictionaries we obtain better performance. In terms of time, the effect of training new dictionaries per frame has also a great impact, obtaining execution times which are 2 to 3 orders of magnitude higher. The application of sparse NBNN to images is specially suitable when dictionaries do not need to be trained per frame, but rather for a set of frames (e.g. a dictionary per scene or shot).

In Figure 9.5 we show close-ups for visual inspection. Our proposed NBNN is able to reconstruct aliased frecuencies where the non-adaptive approach fails to do so (first row in the figure). Overall, the edges obtained are sharper and better preserved with respect the ground truth image.

9.7 Dense Local Training

In this section we provide performance evaluation of our Dense Local Training algorithm [62], where we show substantial quality benefits due to our novel training approach and additional speed-ups caused by the sublinear Spherical Hashing near-

Table 9.5: Performance of ×2, ×3 and ×4 magnification in terms of averaged PSNR (dB) and average execution time (s) on datasets Set5, Set14 and kodak.

		Bicubic		Zeyde et al. [97]		Sparse NBNN	
	S	PSNR	time	PSNR	time	PSNR	time
Set5	2	33.661	0.001	35.677	**4.376**	**35.940**	1032.719
	3	30.392	0.001	31.814	**1.922**	**32.004**	641.534
	4	28.421	0.001	29.612	**1.210**	**29.776**	349.962
Set14	2	30.232	0.002	31.720	**9.244**	**31.947**	1749.697
	3	27.541	0.001	28.585	**4.146**	**28.737**	1175.802
	4	26.000	0.001	26.813	**2.559**	**26.947**	693.179
kodak	2	30.845	0.002	32.133	**16.138**	**32.360**	2541.422
	3	28.426	0.002	29.157	**7.187**	**29.297**	1568.001
	4	27.223	0.002	27.794	**4.551**	**27.888**	1297.318

Table 9.6: Performance of ×2, ×3 and ×4 magnification in terms of averaged IFC and average SSIM on datasets Set5, Set14 and kodak.

		Bicubic		Zeyde et al. [97]		Sparse NBNN	
	S	IFC	SSIM	IFC	SSIM	IFC	SSIM
Set5	2	6.282	0.930	8.111	0.949	**9.573**	**0.951**
	3	3.616	0.869	4.501	0.895	**5.082**	**0.899**
	4	2.342	0.811	2.923	0.841	**3.218**	**0.843**
Set14	2	6.304	0.869	7.871	0.898	**9.108**	**0.903**
	3	3.535	0.774	4.266	0.806	**4.749**	**0.813**
	4	2.259	0.702	2.737	0.732	**3.000**	**0.740**
kodak	2	6.223	0.870	7.602	0.899	**8.659**	**0.904**
	3	3.410	0.779	4.026	0.806	**4.415**	**0.812**
	4	2.126	0.719	2.527	0.742	**2.732**	**0.749**

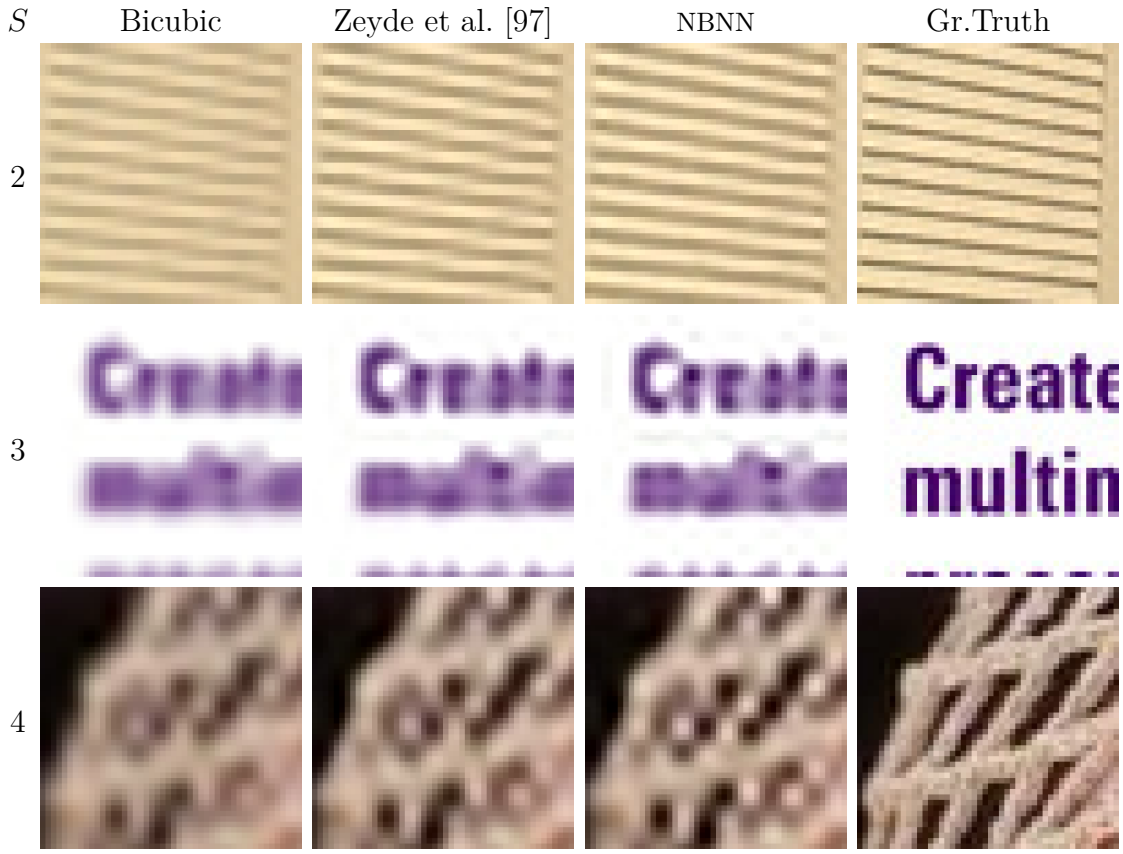


Figure 9.5: Close-ups of the results in Table 9.5 and 9.6 for visual qualitative assessment. Best-viewed zoomed in.

est neighbor search. In this section we assess the impact of each of the contributions, together with the behavior of some of the parameters.

Configuration

DLT is based on ANR [74] and therefore shares some of its parameters. We keep the dictionary size to $d_s = 1024$, however the neighborhood size parameter operates in a completely different way. Previously, ANR selected a reduced set of dictionary atoms in order to build the neighborhoods (i.e. $\mathbf{N_h} \subset \mathbf{D_h}$). The neighborhoods $\mathbf{N_l}$ and $\mathbf{N_h}$ in DLT are constructed directly from a raw pool of patches (i.e. $\mathbf{N_h} \subset \mathbf{X_t}$) of a way greater cardinality (e.g. in the order of hundreds of thousands). In this sense, the neighborhood size is not upper-bounded by the dictionary size and offers a more populated distribution over the manifold, which enables more compact neighborhoods and a higher number of patches within a given distance.

We assess the impact of the neighborhood size in terms of PSNR in Figure 9.6. Smaller dictionaries improve their performance when the neighborhood size is increased. Large neighborhoods have a great impact in the training computational time and memory usage. Obtaining each regressor requires computing the pseudo-inverse $(\mathbf{N}_{\mathbf{l}_j}^\intercal \mathbf{N}_{\mathbf{l}_j} + \lambda \mathbf{I})^{-1}$ from Equation (3.9) which is a square matrix with as many rows as neighbors in the neighborhood. For $d_s = 1024$ we observe small differences from 1000 to 10000 atoms, and thus we select a parameter among the lower end (1300) to stay within reasonable training memory and time consumption.

We show the impact of the number of hyperspheres in terms of PSNR and time in Figure 9.7. We set the number of hyperspheres to be used by the Spherical Hashing NN search to 6 hyperspheres, as this is a reasonable compromise between speed gain and quality loss. In order to also provide the best performance possible, we show as well the performance for 1 hypersphere (i.e. exhaustive search).

Performance

In Table 9.7 and 9.8 we show the objective evaluation (in terms of PSNR, IFC, SSIM and time) of two configurations of DLT (differentiated by a subscript index, e.g. DLT_1) compared to, ANR, GR and the baseline of bicubic interpolation. The performance of our proposed DLT with either configuration clearly outperform the rest of the benchmark for both PSNR, SSIM and time. In terms of PSNR, DLT_1 improves up to 0.6dB the performance over ANR. As of time, our fast SpH sublinear search results in substantial speed-up (between $\times 10$ and $\times 20$ times faster than ANR).

Overall, the performance of the dense training approach together with the spherical hashing search improve greatly both quality and execution time.

In Figure 9.8 we show close-ups for visual inspection. We remark the generally sharper images obtained with our proposed DLT, and the almost complete elimination of ringing artifacts that are present in ANR and GR.

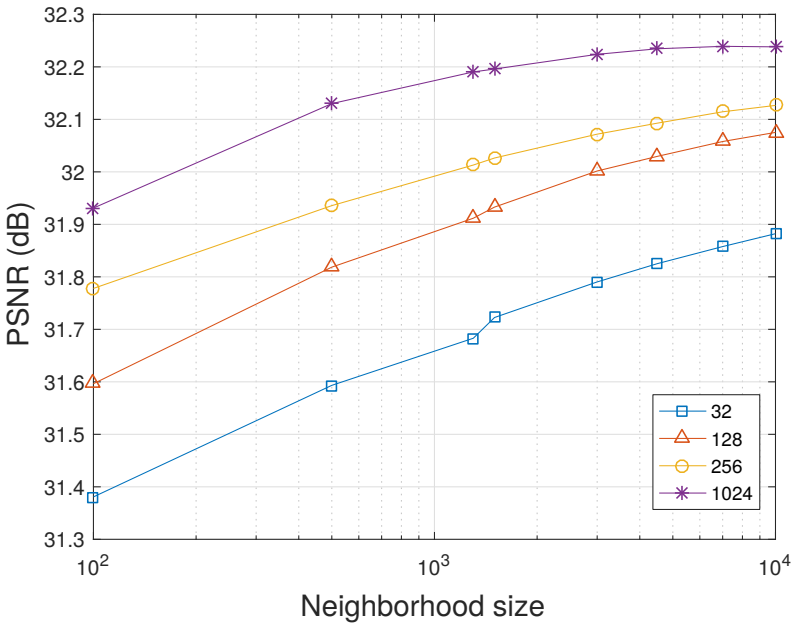
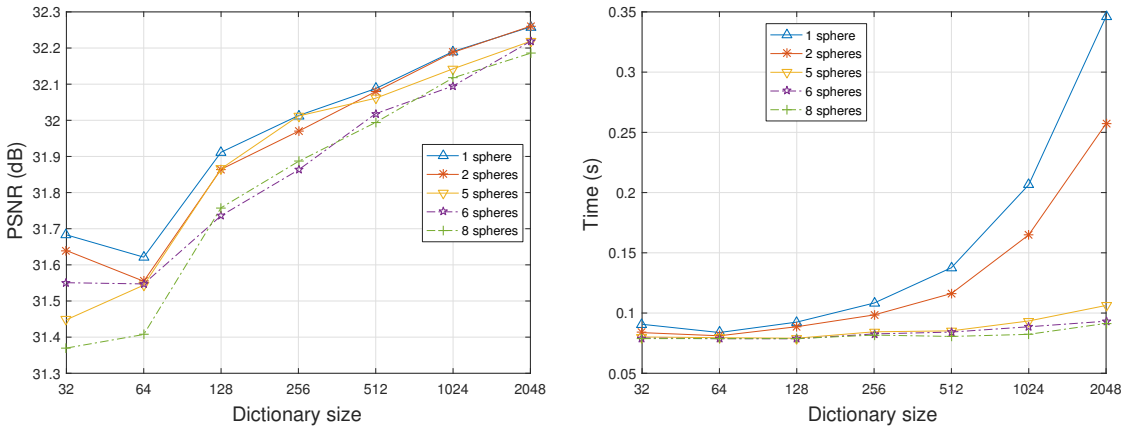


Figure 9.6: DLT parameter configuration: PSNR for different neighborhood and dictionary sizes (32, 128, 256, 1024). For smaller dictionaries, larger neighborhood sizes yield better quality.



(a) (b)

Figure 9.7: DLT parameter configuration: Impact of the dictinary size and the number of spheres selected in the SpH in terms of (a) PSNR and (b) time.

Table 9.7: Performance of ×2, ×3 and ×4 magnification in terms of averaged PSNR (dB) and average execution time (s) on datasets Set5, Set14 and kodak.

		Bicubic		GR		ANR		DLT_1		DLT_6	
	S	PSNR	time	PSNR	time	PSNR	time	PSNR	time	PSNR	time
`Set5`	2	33.661	0.001	35.153	0.515	35.858	0.783	**36.497**	0.099	36.356	**0.045**
	3	30.392	0.001	31.423	0.282	31.926	0.402	**32.542**	0.065	32.407	**0.043**
	4	28.421	0.001	29.342	0.199	29.691	0.273	**30.208**	0.064	30.085	**0.046**
`Set14`	2	30.232	0.002	31.355	1.011	31.801	1.519	**32.190**	0.208	32.095	**0.088**
	3	27.541	0.001	28.305	0.568	28.647	0.801	**29.082**	0.136	29.016	**0.085**
	4	26.000	0.001	26.589	0.421	26.846	0.564	**27.317**	0.121	27.209	**0.085**
`kodak`	2	30.845	0.002	31.873	1.711	32.245	2.573	**32.647**	0.328	32.558	**0.150**
	3	28.426	0.002	28.987	0.968	29.213	1.395	**29.567**	0.225	29.512	**0.142**
	4	27.223	0.002	27.639	0.716	27.806	0.975	**28.069**	0.199	28.021	**0.143**

Table 9.8: Performance of ×2, ×3 and ×4 magnification in terms of averaged IFC and average SSIM on datasets Set5, Set14 and kodak.

		Bicubic		GR		ANR		DLT_1		DLT_6	
	S	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM
`Set5`	2	6.282	0.930	**8.663**	0.945	8.523	0.950	8.617	**0.954**	8.511	0.954
	3	3.616	0.869	4.649	0.885	4.689	0.897	**4.880**	**0.909**	4.772	0.906
	4	2.342	0.811	2.987	0.828	3.030	0.842	**3.192**	**0.860**	3.103	0.854
`Set14`	2	6.304	0.869	**8.433**	0.898	8.193	0.901	8.173	**0.905**	8.106	0.904
	3	3.535	0.774	4.448	0.803	4.415	0.810	**4.514**	**0.819**	4.442	0.817
	4	2.259	0.702	2.822	0.728	2.829	0.735	**2.925**	**0.749**	2.863	0.746
`kodak`	2	6.223	0.870	**8.168**	0.899	7.885	0.902	7.783	**0.908**	7.729	0.907
	3	3.410	0.779	4.195	0.804	4.145	0.809	**4.199**	**0.819**	4.145	0.817
	4	2.126	0.719	2.599	0.740	2.589	0.744	**2.652**	**0.754**	2.603	0.752

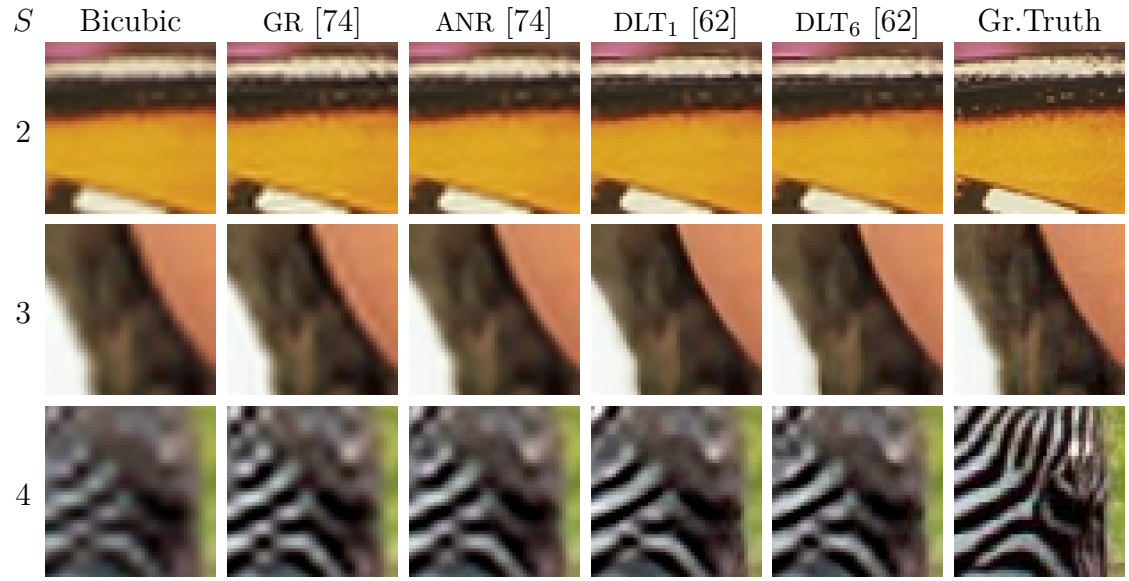


Figure 9.8: Close-ups of the results in Table 9.7 and 9.8 for visual qualitative assessment. Best-viewed zoomed in.

9.8 Half Hypersphere Confinement

The HHC SR is an extension of our previous DLT where we take advantage of antipodally invariance both during training and testing stages.

Configuration

HHC utilizes the same parameters as DLT, as it is essentially the same algorithm with support for antipodal invariance. We perform several experiments in order to assess the best configuration in terms of neighborhood sizes and number of hyperspheres, and we discuss why it is a good idea to increase the dictionary size and the number of spheres, showing some experiments where the benefits are outlined. In Figure 9.9 we show the impact of the neighborhood size for several dictionary sizes, where we observe similar behavior as in DLT. Smaller dictionaries require larger neighborhoods, and there is a saturation point where the quality stabilizes. We fix the neihghborhood size to 4250 as quality improvement has already stalled.

The dictionary size d_s is not necessarily associated with the number of hyperspheres used during testing time. Our hashing scheme defines several hash codes or buckets, and the regressors are labeled with them during training time. In the case of having more than one regressor per bucket, a reranking strategy is followed and thus the best-suited regressor is obtained from the bucket's candidates. The ratio between the number of sparse atoms and the number of buckets (2^s where s is the number of hyperspheres) gives an average number of regressors per hash code.

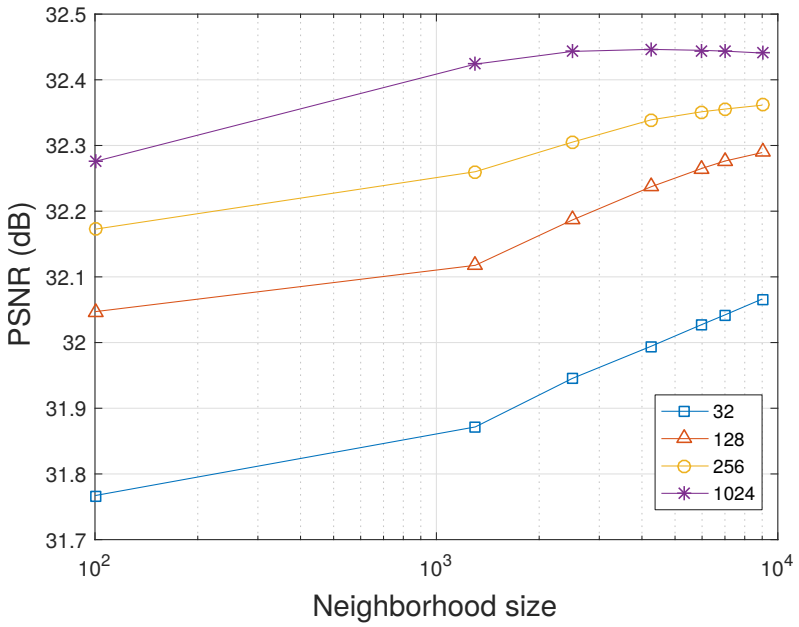
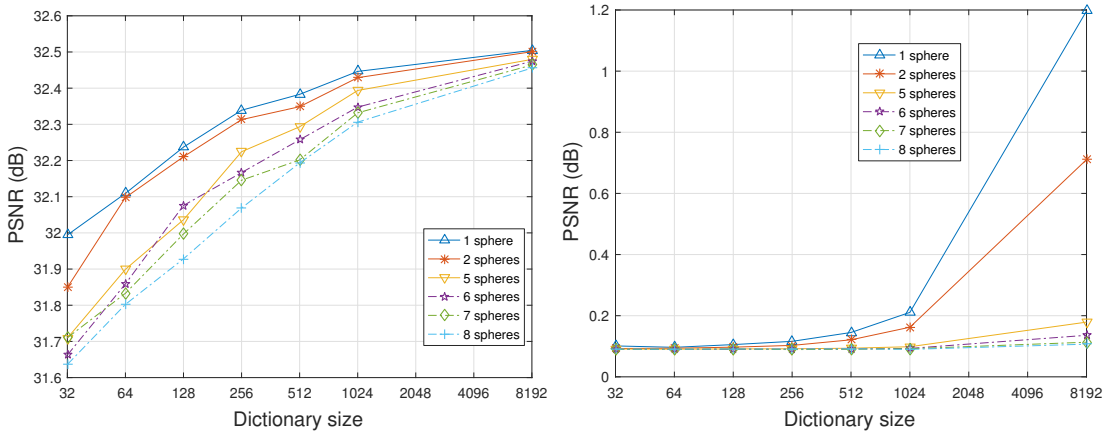


Figure 9.9: HHC parameter configuration: PSNR for different neighborhood and dictionary sizes (32, 128, 256, 1024).

In Figure 9.10 we show that our algorithm scales well in terms of quality when increasing the size of the sparse dictionary, and therefore, is worth increasing its size and adapting the number of hyperspheres to obtain the desired quality and speed trade-off. We aim to at the same execution time as DLT used with 1024 atoms and 6 hyperspheres. We obtain a very similar time figure (while obtaining substantially improved PSNR quality) with 7 spheres and 8192 atoms. In Figure 9.11 we show how our algorithm scales better by increasing the dictionary size than A^+, which improvement is always smaller and tends to saturate earlier. We obtain maximum quality by setting our dictionary size to 8192 elements and, afterwards, fixing a number of hyperspheres which gives us the desired speed.

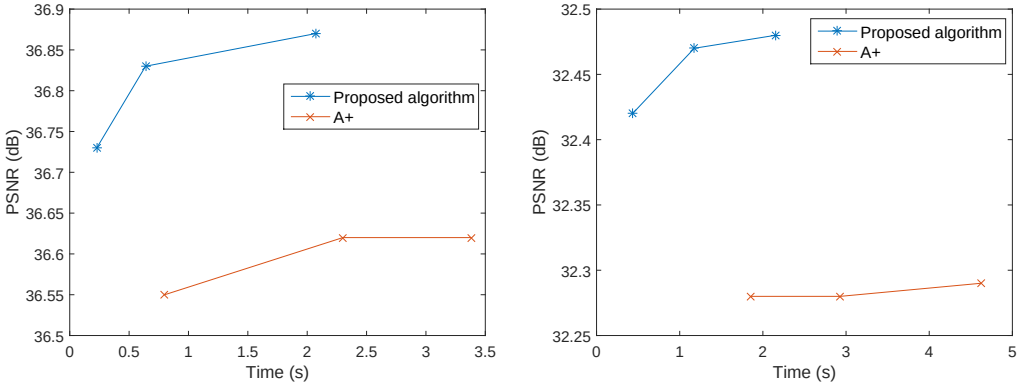
Note also that even without enlarging the dictionary we obtain better quality performance. In Figure 9.11 we show that our methods is consistently obtaining better PSNR values (about 0.2dB higher) for different dictionary sizes, and that even with 1024 atoms we perform better than A^+ with 8192 atoms.

We present two configurations of HHC in the evaluation tables: 1 hypersphere (i.e. exhaustive search) which sets an upper quality limit and 7 hyperspheres (differentiated with a numerical subscript, e.g. HHC_7) which is our optimal configuration in terms of quality vs speed trade-off. By showing both configurations we evaluate the



(a) (b)

Figure 9.10: HHC parameter configuration: Impact of the dictinary size and the number of spheres selected in the SpH in terms of (a) PSNR and (b) time.



(a) Set5 (b) Set14

Figure 9.11: HHC parameter configuration: PSNR vs time values for dictionary sizes of 1024, 4096 and 8192 atoms (from left to right in the lines) for HHC (1 hypersphere) and A^+ .

Table 9.9: Performance of ×2, ×3 and ×4 magnification in terms of averaged PSNR (dB) and average execution time (s) on datasets Set5, Set14 and kodak.

		Bicubic		A^+		DLT_6		HHC_1		HHC_7	
	S	PSNR	time	PSNR	time	PSNR	time	PSNR	time	PSNR	time
`Set5`	2	33.661	0.001	36.571	0.823	36.356	0.045	**36.905**	0.896	36.836	**0.076**
	3	30.392	0.001	32.597	0.396	32.407	0.043	**32.814**	0.430	32.768	**0.062**
	4	28.421	0.001	30.285	0.279	30.085	0.046	**30.481**	0.280	30.442	**0.069**
`Set14`	2	30.232	0.002	32.283	1.585	32.095	0.088	**32.504**	1.790	32.464	**0.146**
	3	27.541	0.001	29.127	0.840	29.016	0.085	**29.271**	0.865	29.233	**0.120**
	4	26.000	0.001	27.314	0.586	27.209	0.085	**27.451**	0.544	27.415	**0.120**
`kodak`	2	30.845	0.002	32.721	2.722	32.558	0.150	**32.908**	3.091	32.866	**0.249**
	3	28.426	0.002	29.579	1.437	29.512	0.142	**29.697**	1.484	29.664	**0.205**
	4	27.223	0.002	28.104	1.006	28.021	0.143	**28.186**	0.933	28.173	**0.190**

effect of the approximate search both in quality drop and in time speed-up, showing at the same time the full potential of the antipodal search and GIBP features.

For the comparison, A^+ [76] uses a dictionary of 1024 atoms and a neighborhood size of 2048 atoms, as setting it to 4250 degraded their quality results. The rest of the parameters are set equally for all the methods.

Performance

We show objective evaluation of all methods in Table 9.9 (PSNRand time) and Table 9.10 (IFC and SSIM). First of all, the PSNR obtained with our HHC SR is consistently around 0.2dB higher than that of A^+, which is the most related compared method. The speed-up with respect A+ ranges from ×4.6 to 9.3. We are consistently the best-performers both in SSIM and IFC for all datasets and magnification factors, confirming the good performance of our method.

Secondly, the algorithmic speed of the spherical hashing approach (i.e. comparison between $s = 1$ and $s = 7$) ranges from ×4.8 to 11 depending on the upscaling factors. The drop in quality is very reduced and ranges from 0.01 to 0.07dB. With $s = 7$ we clearly outperform methods in running time (with the exception of bicubic) while being highly competitive in quality (PSNR, IFC, SSIM).

In Figure 9.12 we show close-ups for visual inspection. In the first row we show how our HHC_1 is able to better reconstruct complex structures (note that the continuity and structure of the diagonal lines is recovered). In the second row we show an example where there is aliasing present, and how we can better neutralize it. In the third row we show an example where images upscaled by HHC SR have sharper edges and less of artifacts.

Table 9.10: Performance of ×2, ×3 and ×4 magnification in terms of averaged IFC and average SSIM on datasets Set5, Set14 and kodak.

		Bicubic		A^+		DLT_6		HHC_1		HHC_7	
	S	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM
Set5	2	6.282	0.930	9.031	0.955	8.511	0.954	**9.300**	**0.957**	9.231	0.956
	3	3.616	0.869	5.036	0.909	4.772	0.906	**5.192**	**0.913**	5.148	0.912
	4	2.342	0.811	3.277	0.861	3.103	0.854	**3.389**	**0.866**	3.356	0.865
Set14	2	6.304	0.869	8.551	0.906	8.106	0.904	**8.768**	**0.909**	8.712	0.908
	3	3.535	0.774	4.644	0.819	4.442	0.817	**4.775**	**0.823**	4.736	0.822
	4	2.259	0.702	3.002	0.749	2.863	0.746	**3.088**	**0.754**	3.058	0.753
kodak	2	6.223	0.870	8.117	0.908	7.729	0.907	**8.283**	**0.911**	8.248	0.910
	3	3.410	0.779	4.301	0.818	4.145	0.817	**4.405**	**0.822**	4.374	0.821
	4	2.126	0.719	2.718	0.754	2.603	0.752	**2.786**	**0.758**	2.765	0.757

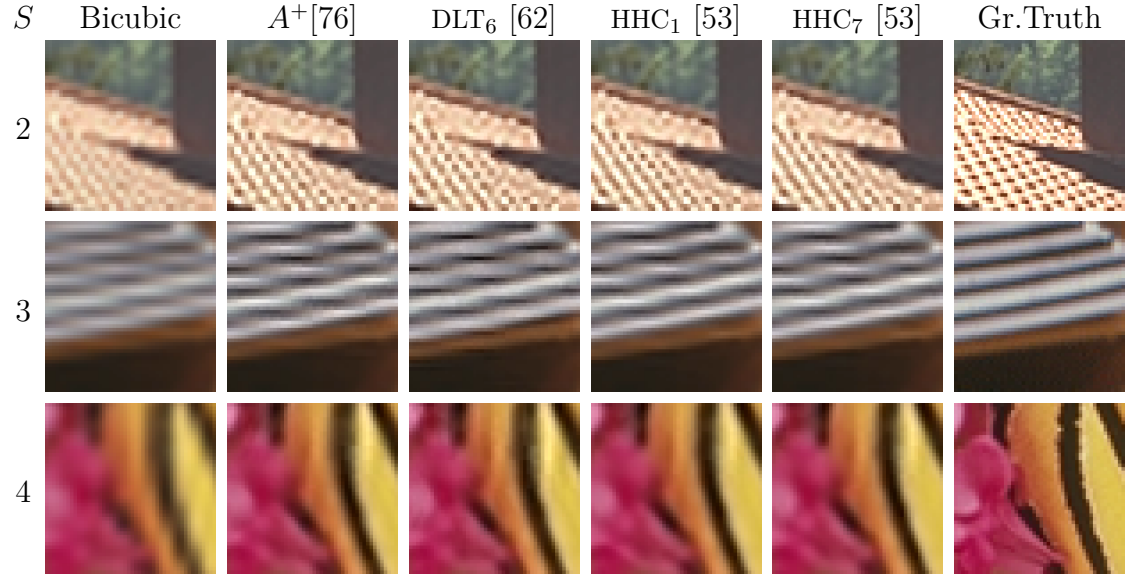


Figure 9.12: Close-ups of the results in Table 9.9 and 9.10 for visual qualitative assessment. Best-viewed zoomed in.

9.9 Naive Bayes SR Forest

Configuration

For our NBSRF the two most relevant parameters are the number of trees in the forest, and the depth of each of the trees. In Figure 9.13 we show the impact of those parameters in terms of PSNR and time. We see that the computational cost of adding more trees is linear, but with a small slope (regardless of the number of trees, only one regression takes place). The bottom left chart also serves to validate the Local Naive Bayes tree selection algorithm. The addition of more trees to the ensemble consistently produces better accuracy, even though just one of them is actually used to infer the appearance correction for the coarse patch. We set the number of trees of our forest to 8 as even larger number of trees does not seem necessary given the saturation of the curve. We fix the tree depth to 11, which translates into 2048 leaf nodes per tree. In figure bottom right we see that increasing this number even further results in increased quality. The rationale behind limiting our trees to 2048 leaves goes in the direction of staying within reasonable memory model size (i.e. the model size grows linearly with the number of trees and exponentially with the tree depth).

Performance

In Table 9.7 and 9.8 we show the objective evaluation (in terms of PSNR, IFC, SSIM and time) of our proposed NBSRF compared with DLT_6 which also uses sublinear search and A^+ which uses exhaustive search.

NBSRF is the best performer in terms of PSNR, SSIM and IFC, with improvements around 0.2dB with respect to A^+. NBSRF is reasonably fast, however is not the fastest within the compared algorithms as DLT also relies in a sublinear search and does not require traversing several binary structures (i.e. tree ensemble).

In Figure 9.14 we show some close-ups for subjective evaluation. NBSRF shows an outstanding robustness against ringing and aliasing effects.

9.10 Patch Symmetry Collapse

Configuration

In Figure 9.15a and 9.15b we show the behavior of the most important two parameters to be selected in our PSyCo SRalgorithm. The neighborhood size has higher impact and its optimal value increases for smaller dictionary sizes (as each neighborhood covers more span within the manifold). The regularization weighting term λ has a lesser impact and its optimal value increases for big neighborhoods and small

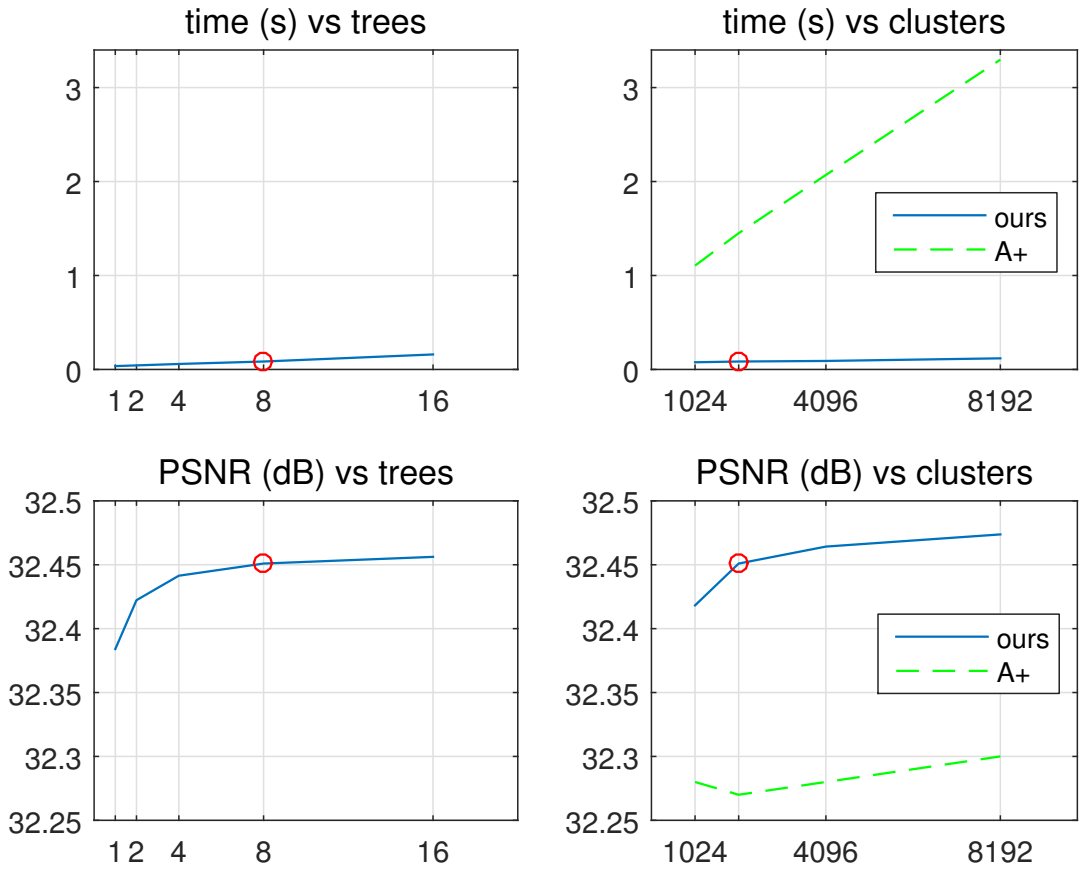


Figure 9.13: NBSRF parameter configuration: PSNR and time for several number of trees (left) and leaf nodes per tree (right). The reference configuration is marked in red.

Table 9.11: Performance of ×2, ×3 and ×4 magnification in terms of averaged PSNR (dB) and average execution time (s) on datasets Set5, Set14 and kodak.

		Bicubic		A^+		DLT_6		NBSRF [65]	
	S	PSNR	time	PSNR	time	PSNR	time	PSNR	time
Set5	2	33.661	0.001	36.571	0.823	36.356	**0.045**	**36.757**	**0.045**
	3	30.392	0.001	32.597	0.396	32.407	**0.043**	**32.741**	0.083
	4	28.421	0.001	30.285	0.279	30.085	**0.046**	**30.430**	0.124
Set14	2	30.232	0.002	32.283	1.585	32.095	**0.088**	**32.453**	0.089
	3	27.541	0.001	29.127	0.840	29.016	**0.085**	**29.252**	0.123
	4	26.000	0.001	27.314	0.586	27.209	**0.085**	**27.415**	0.180
kodak	2	30.845	0.002	32.721	2.722	32.558	0.150	**32.805**	**0.139**
	3	28.426	0.002	29.579	1.437	29.512	**0.142**	**29.626**	0.213
	4	27.223	0.002	28.104	1.006	28.021	**0.143**	**28.170**	0.289

Table 9.12: Performance of ×2, ×3 and ×4 magnification in terms of averaged IFC and average SSIM on datasets Set5, Set14 and kodak.

		Bicubic		A^+		DLT_6		NBSRF [65]	
	S	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM
Set5	2	6.282	0.930	9.031	**0.955**	8.511	0.954	**9.124**	**0.955**
	3	3.616	0.869	5.036	0.909	4.772	0.906	**5.084**	**0.910**
	4	2.342	0.811	3.277	0.861	3.103	0.854	**3.311**	**0.863**
Set14	2	6.304	0.869	8.551	0.906	8.106	0.904	**8.643**	**0.907**
	3	3.535	0.774	4.644	0.819	4.442	0.817	**4.680**	**0.821**
	4	2.259	0.702	3.002	0.749	2.863	0.746	**3.013**	**0.751**
kodak	2	6.223	0.870	8.117	0.908	7.729	0.907	**8.207**	**0.909**
	3	3.410	0.779	4.301	0.818	4.145	0.817	**4.327**	**0.819**
	4	2.126	0.719	2.718	0.754	2.603	0.752	**2.723**	**0.756**

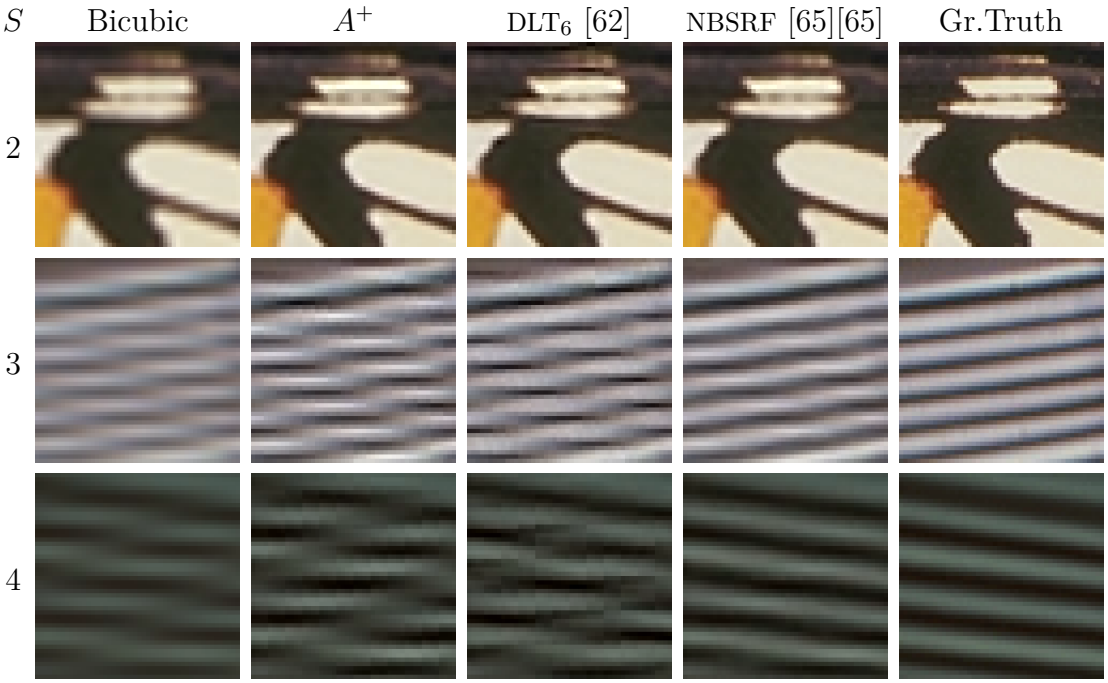


Figure 9.14: Close-ups of the results in Table 9.11 and 9.12 for visual qualitative assessment. Best-viewed zoomed in.

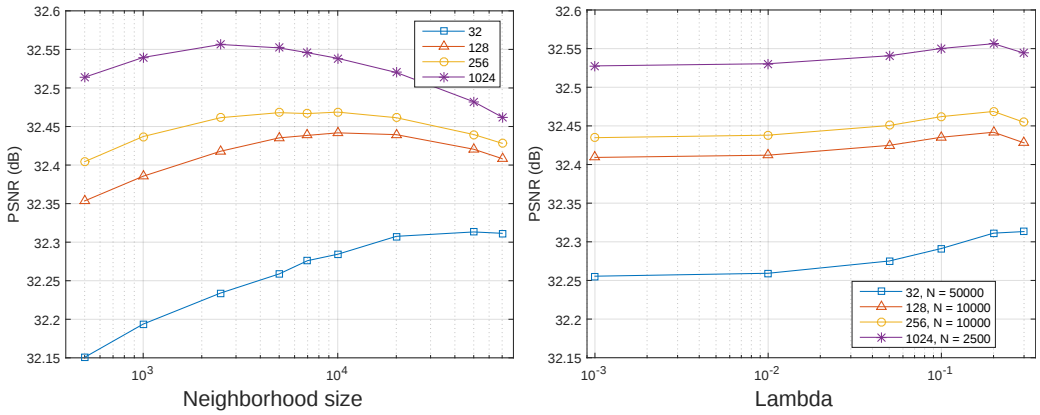


Figure 9.15: PSyCo parameter configurations: (a) shows the impact of the neighborhood size and (b) the impact of the regularization weighting term λ.

dictionaries. We also note that $\lambda = 0.2$ is a good compromise across all possible configurations, and thus we recommend its usage for a first approach when optimizing the neighborhood size. We note that fine tuning over lambda is notable faster when using Frobenius norm as the dimensionality of the matrix inversion $(\overset{\circ}{\mathbf{C}}_i \overset{\circ}{\mathbf{C}}_i{}^\intercal + \lambda \mathbf{I})^{-1}$ in Equation (8.5) does not grow with the neighborhood size, but rather is fixed to the feature dimensionality.

We present two different configurations, with 32 and 1024 atoms. For the first one, we set a neighborhood size of 42000 and $\lambda = 0.25$; for the second one the neighborhood size is set to 2750 and $\lambda = 0.18$.

Performance

9.11 Benchmarking

In this section we benchmark the algorithms presented in this thesis together with some of the current SR state of the art. The motivation for such a benchmark is to focus less in each algorithm individually (and its configuration and parameters, both of them discussed previously) and rather see in a more explicit way the overall evolution line, and how does PSyCo, the latest and most mature algorithm in this dissertation, compare with other methods of the state of the art.

The methods that we include in our benchmark are: The current Super Resolution using Convolutional Neuronal Networks (SRCNN) deep learning method presented by Dong et al. [18] with their recommended 9-5-5 network (note the superior performance when compared to the 9-1-5 network of their earlier publication [17]), the A^+ anchored regression algorithm of Timofte et al. [76], the recently published SR

Table 9.13: Performance of ×2, ×3 and ×4 magnification in terms of averaged PSNR (dB) and average execution time (s) on datasets Set5, Set14 and kodak.

		Bicubic		A^+		PSyCo$_{32}$		PSyCo$_{1024}$	
	S	PSNR	time	PSNR	time	PSNR	time	PSNR	time
Set5	2	33.661	0.001	36.571	0.823	36.598	**0.027**	**36.905**	0.091
	3	30.392	0.001	32.597	0.396	32.636	**0.040**	**32.934**	0.098
	4	28.421	0.001	30.285	0.279	30.326	**0.054**	**30.627**	0.106
Set14	2	30.232	0.002	32.283	1.585	32.321	**0.043**	**32.554**	0.191
	3	27.541	0.001	29.127	0.840	29.127	**0.065**	**29.360**	0.190
	4	26.000	0.001	27.314	0.586	27.299	**0.091**	**27.566**	0.201
kodak	2	30.845	0.002	32.721	2.722	32.657	**0.071**	**32.902**	0.317
	3	28.426	0.002	29.579	1.437	29.574	**0.107**	**29.744**	0.312
	4	27.223	0.002	28.104	1.006	28.070	**0.162**	**28.284**	0.346

Table 9.14: Performance of ×2, ×3 and ×4 magnification in terms of averaged IFC and average SSIM on datasets Set5, Set14 and kodak.

		Bicubic		A^+		PSyCo$_{32}$		PSyCo$_{1024}$	
	S	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM
Set5	2	6.282	0.930	9.031	0.955	9.114	0.955	**9.239**	**0.957**
	3	3.616	0.869	5.036	0.909	5.088	0.910	**5.206**	**0.914**
	4	2.342	0.811	3.277	0.861	3.312	0.861	**3.413**	**0.870**
Set14	2	6.304	0.869	8.551	0.906	8.640	0.906	**8.735**	**0.909**
	3	3.535	0.774	4.644	0.819	4.683	0.819	**4.780**	**0.824**
	4	2.259	0.702	3.002	0.749	3.031	0.749	**3.101**	**0.757**
kodak	2	6.223	0.870	8.117	0.908	8.219	0.908	**8.267**	**0.911**
	3	3.410	0.779	4.301	0.818	4.340	0.818	**4.399**	**0.822**
	4	2.126	0.719	2.718	0.754	2.735	0.754	**2.795**	**0.760**

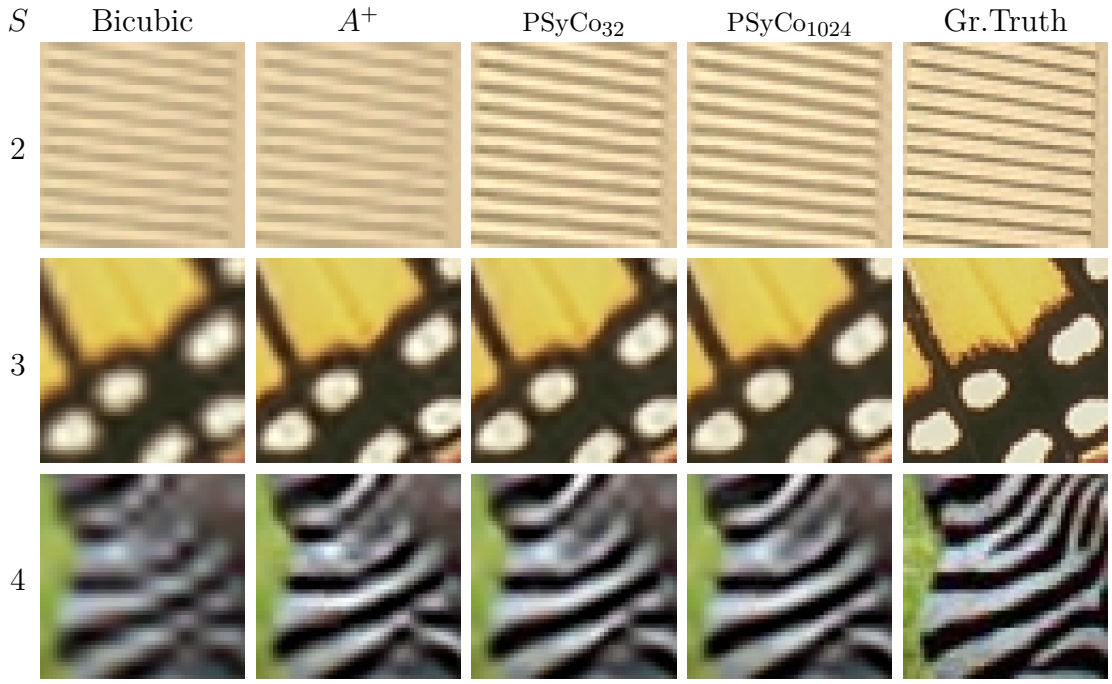


Figure 9.16: Close-ups of the results in Table 9.13 and 9.14 for visual qualitative assessment. Best-viewed zoomed in.

forest with alternative training ASRF of Schulter et al. [68] and the Transformed Self-Exemplars Single-Image SR of Huang et al. [35].

We train A^+, ASRF and *PSyCo* with the same 91 images provided by Yang et al. in their sparse coding SR [95]. As for SRCNN, we use the network provided by their authors which has been trained with the ImageNet dataset (in the order of hundred thousand images) [14].

For the compared methods we used the code publicly available from the author's website. Our code is a MATLAB + MEX implementation with parallel support.

In Table 9.15 and 9.16 we show the averaged PSNR, IFC, SSIM indices and execution times of the benchmark. PSyCo with 1024 atoms obtains the best PSNR values, around 0.3dB higher across all s and datasets when compared to the most related algorithm A^+. We also outperform the most competitive methods (SRCNN and Super-Resolution Forest with alternative training (ASRF)) in PSNR by up to 0.3dB. In terms of time, both our configurations are the fastest of the benchmark, specially our proposed (32), which is an order of magnitude faster than any other method. We also note that our methods are trained in less than two hours, which contrasts with the SRCNN method trained with ImageNet. The measured IFC values are consistently the highest among the benchmark, and we highlight the fact that for most magnification factors, PSyCo with 32 atoms obtains the runner-up IFC, confirming the good performance of our time- and memory-effective configuration. In

Table 9.15: Performance of ×2, ×3 and ×4 magnification in terms of averaged PSNR (dB) and execution time (s) on datasets Set5, Set14 and Kodak. Best results in bold and runner-up in blue.

		Bicubic		SRCNN [18]		TSelfEx [35]		ASRF [68]		A+ [76]		PSyCo (32)		PSyCo (1024)	
	s	PSNR	time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
Set5	2	33.66	0.001	36.66	4.722	36.50	42.521	36.69	1.278	36.57	0.823	36.60	**0.027**	**36.91**	0.091
	3	30.39	0.001	32.75	5.226	36.62	31.008	32.57	1.026	32.60	0.396	32.64	**0.040**	**32.93**	0.098
	4	28.42	0.001	30.48	9.962	30.33	26.728	30.20	1.071	30.29	0.279	30.33	**0.054**	**30.63**	0.106
Set14	2	30.23	0.002	32.45	8.204	32.23	98.645	32.36	2.134	32.28	1.585	32.32	**0.043**	**32.55**	0.191
	3	27.54	0.001	29.29	8.098	29.16	70.176	29.12	1.674	29.13	0.840	29.13	**0.065**	**29.36**	0.190
	4	26.00	0.001	27.50	8.305	27.40	63.873	27.31	1.386	27.31	0.586	27.30	**0.091**	**27.57**	0.201
kodak	2	30.85	0.002	32.81	14.367	32.65	195.70	32.76	3.360	32.72	2.722	32.66	**0.071**	**32.90**	0.317
	3	28.43	0.002	29.65	15.026	29.52	135.243	29.63	2.555	29.58	1.437	29.57	**0.107**	29.74	0.312
	4	27.22	0.002	28.17	14.069	28.14	115.652	28.17	2.204	28.10	1.006	28.07	**0.162**	28.28	0.346

Table 9.16: Performance of ×2, ×3 and ×4 magnification in terms of averaged IFC and SSIM on datasets Set5, Set14 and Kodak. Best results in bold and runner-up in blue.

		Bicubic		SRCNN [18]		TSelfEx [35]		ASRF [68]		A+ [76]		PSyCo (32)		PSyCo (1024)	
	s	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM	IFC	SSIM
Set5	2	6.28	0.930	8.04	0.954	7.81	0.930	8.56	0.950	9.03	0.955	9.11	0.955	**9.24**	0.957
	3	3.62	0.869	4.66	0.909	4.75	0.868	4.93	0.908	5.04	0.909	5.09	0.910	**5.21**	0.914
	4	2.34	0.811	2.99	0.863	3.17	0.810	3.19	0.857	3.28	0.861	3.31	0.861	**3.41**	0.870
Set14	2	6.30	0.869	7.78	0.907	7.59	0.869	8.18	0.906	8.55	0.906	8.64	0.906	**8.74**	0.909
	3	3.54	0.774	4.34	0.821	4.37	0.774	4.53	0.818	4.64	0.819	4.68	0.819	**4.78**	0.824
	4	2.26	0.702	2.75	0.751	2.89	0.702	2.92	0.746	3.00	0.749	3.03	0.749	**3.10**	0.757
kodak	2	6.22	0.870	7.15	0.907	6.78	0.869	7.39	0.9070	8.12	0.908	8.22	0.908	**8.27**	0.911
	3	3.41	0.779	3.90	0.818	3.81	0.778	4.03	0.8150	4.30	0.818	4.34	0.818	**4.40**	0.822
	4	2.13	0.719	2.42	0.754	2.46	0.719	2.53	0.7510	2.72	0.754	2.74	0.754	**2.80**	0.760

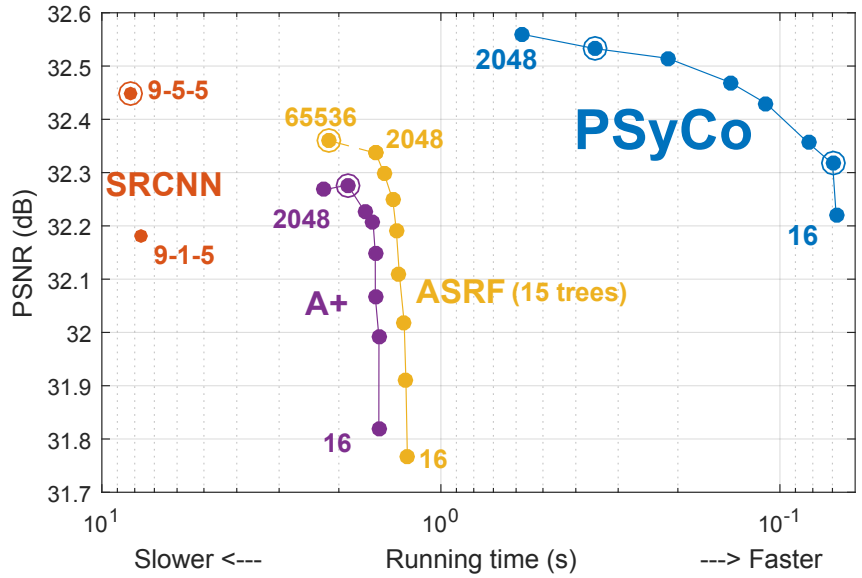


Figure 9.17: PSNR vs time (s) of PSyCo SR compared to other SR methods for dictionary sizes from 16 to 2048, in power-of-two increments. Experiment run on Set14 and $\times 2$ magnification factor. Circled points are found in Table 9.15.

Figures 9.18, 9.19 and 9.20 we show some close-ups for subjective evaluation of $\times 2$, $\times 3$ and $\times 4$ upscaling factors respectively. We highlight the generally sharper edges, the less proliferation of ringing artifacts and the resilience to aliasing which results in better preserved structures. For those methods which depend on a dictionary, we test a $\times 2$ upscaling factor on `Set14` for several dictionary sizes and measure PSNR and times (see Figure 9.17) to compare performances for equal dictionary sizes.

In Figure 9.21 we show the evolution of our methods in terms of the three-fold defining factors of SR: Memory efficiency (represented as a the memory compression rate with respect to ANR), speed (in frames per second) and quality (in terms of PSNR). The triangle representing ANR is widely surpassed in the three metrics of the radar plot. Interestingly, there is a clear progression in terms of quality and speed for algorithms such as DLT, HHC and NBSRF. However, the model size of such algorithms does not improve, as both HHC and NBSRF have substantially bigger models (e.g. more regressors, more anchor points) than ANR. The benefits of the PSyCo are obvious: With the compact configuration of 32 atoms, it increases greatly in speed and memory efficiency, while still retaining a very competitive quality performance, standing clearly out as the method with a higher score in this three dimensional assesment.

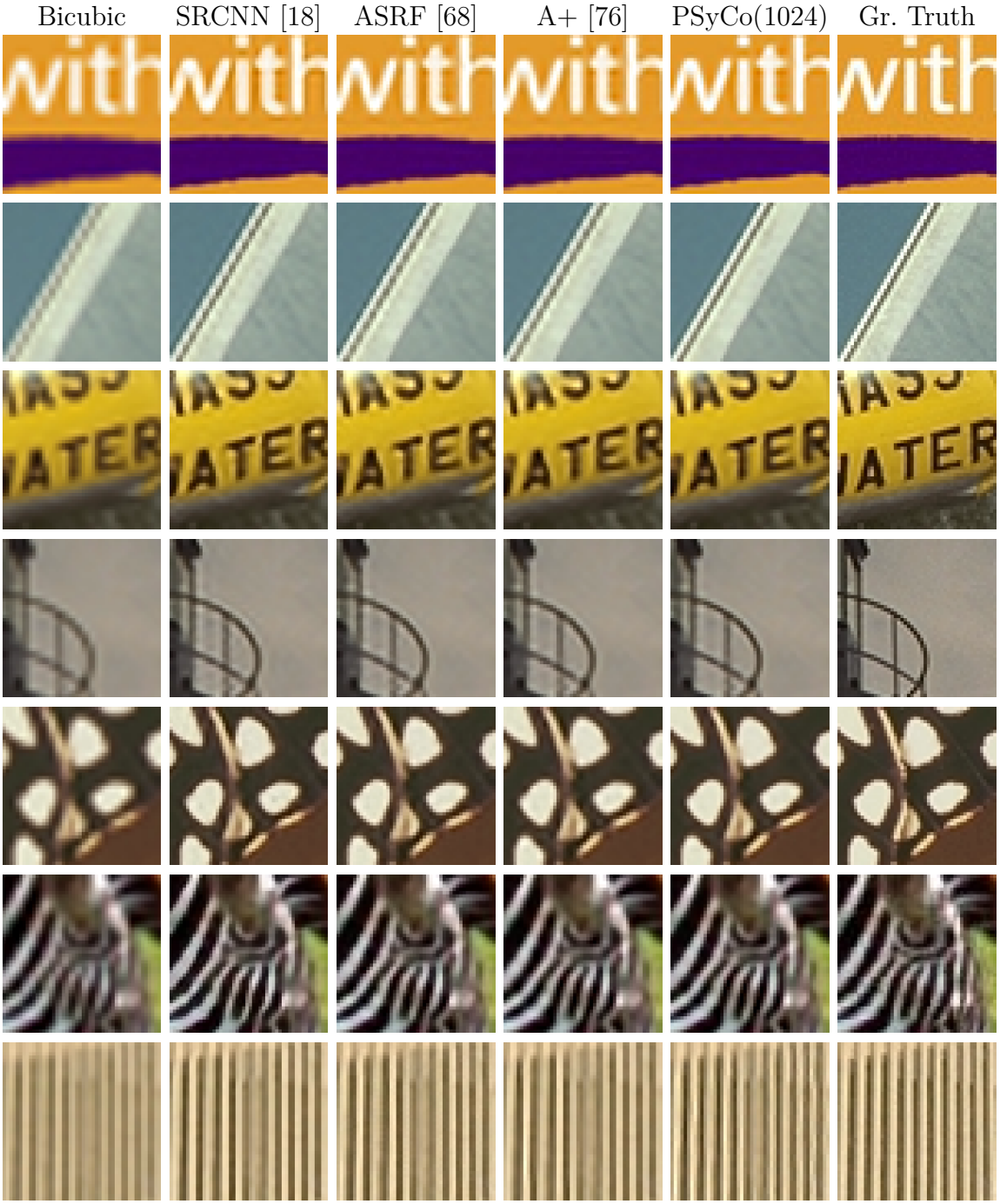


Figure 9.18: Close-ups of the results for visual qualitative assessment of a $\times 2$ magnification factor from the datasets in the benchmark. Best-viewed zoomed in.

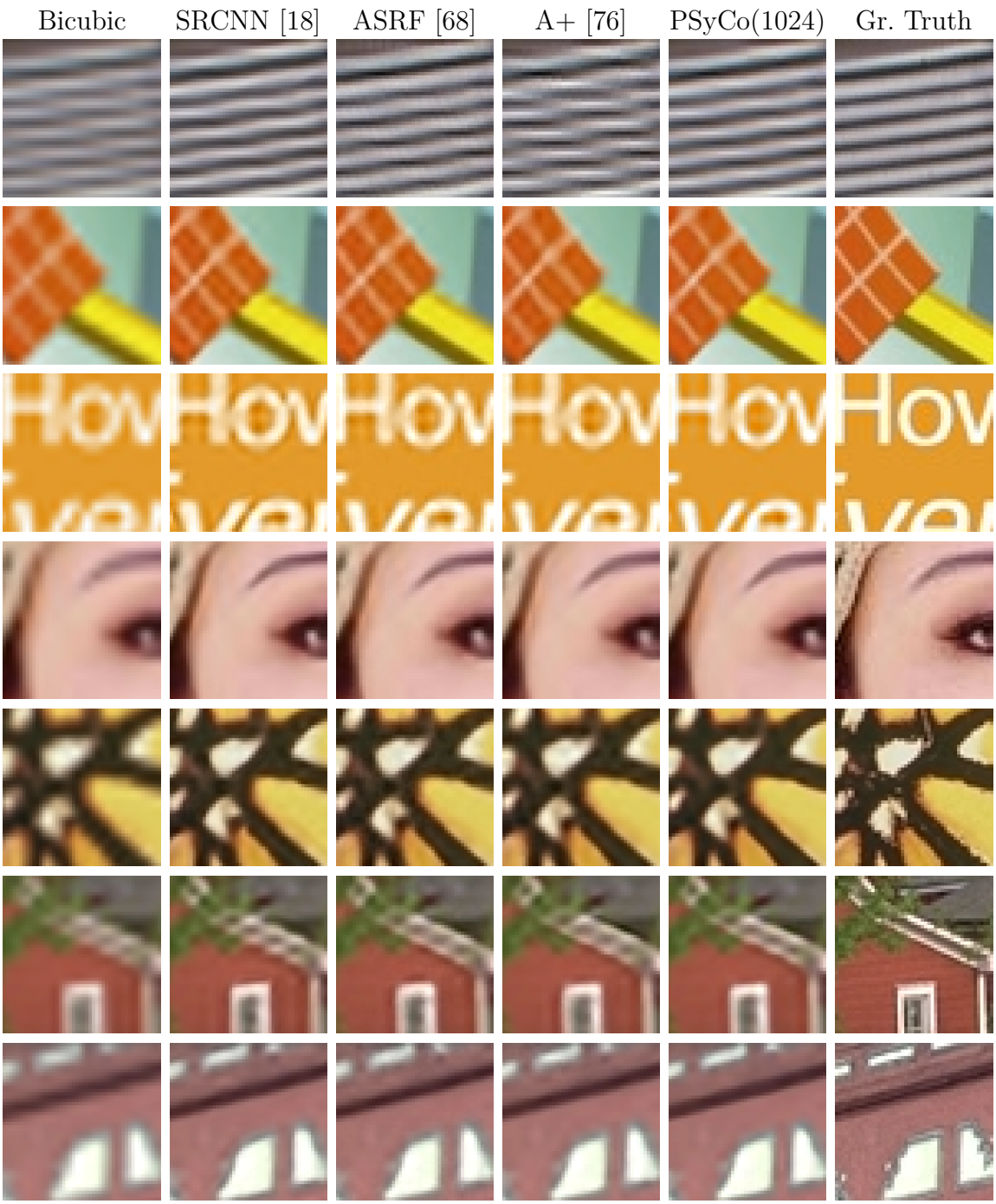


Figure 9.19: Close-ups of the results for visual qualitative assessment of a $\times 3$ magnification factor from the datasets in the benchmark. Best-viewed zoomed in.

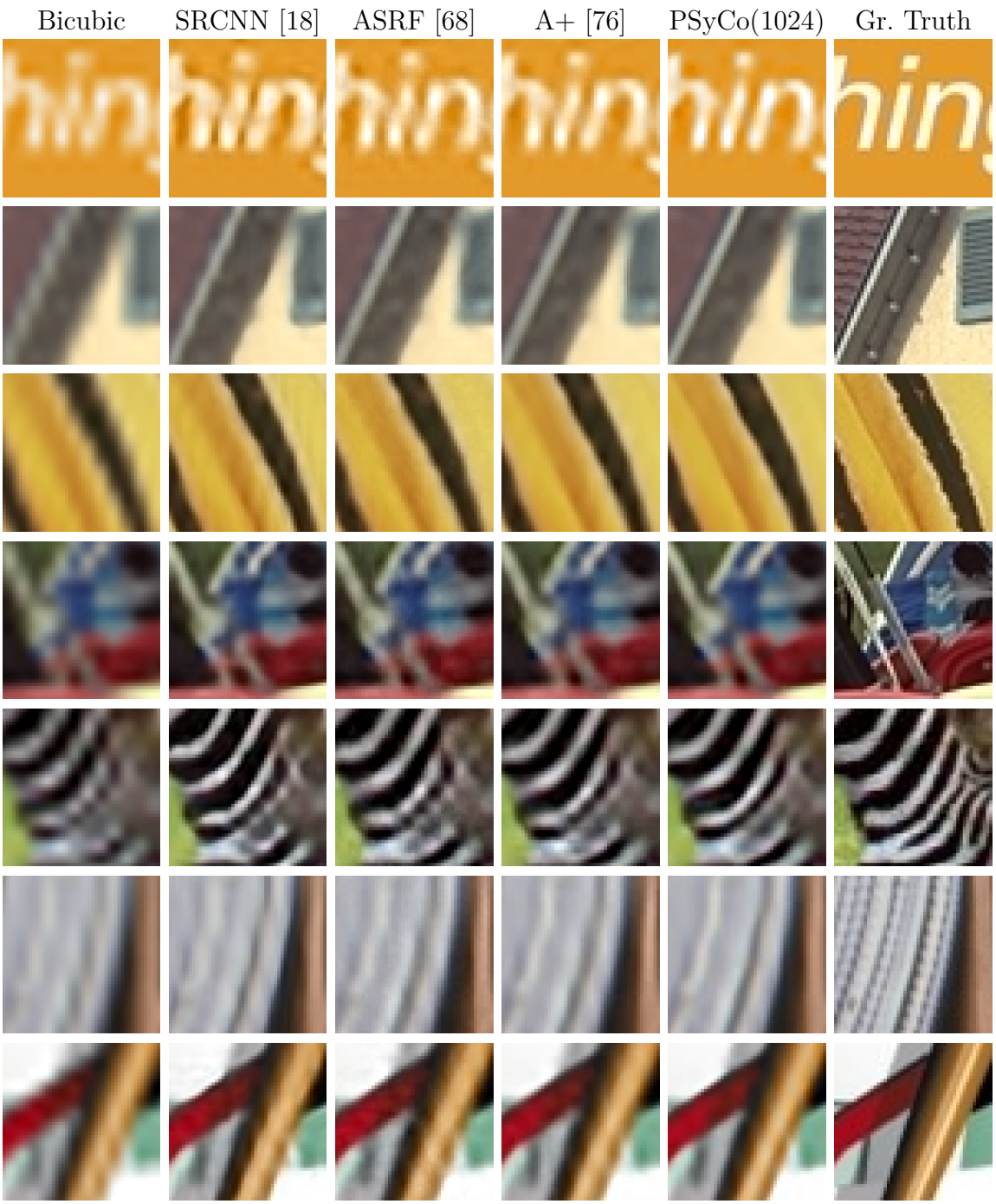


Figure 9.20: Close-ups of the results for visual qualitative assessment of a $\times 4$ magnification factor from the datasets in the benchmark. Best-viewed zoomed in.

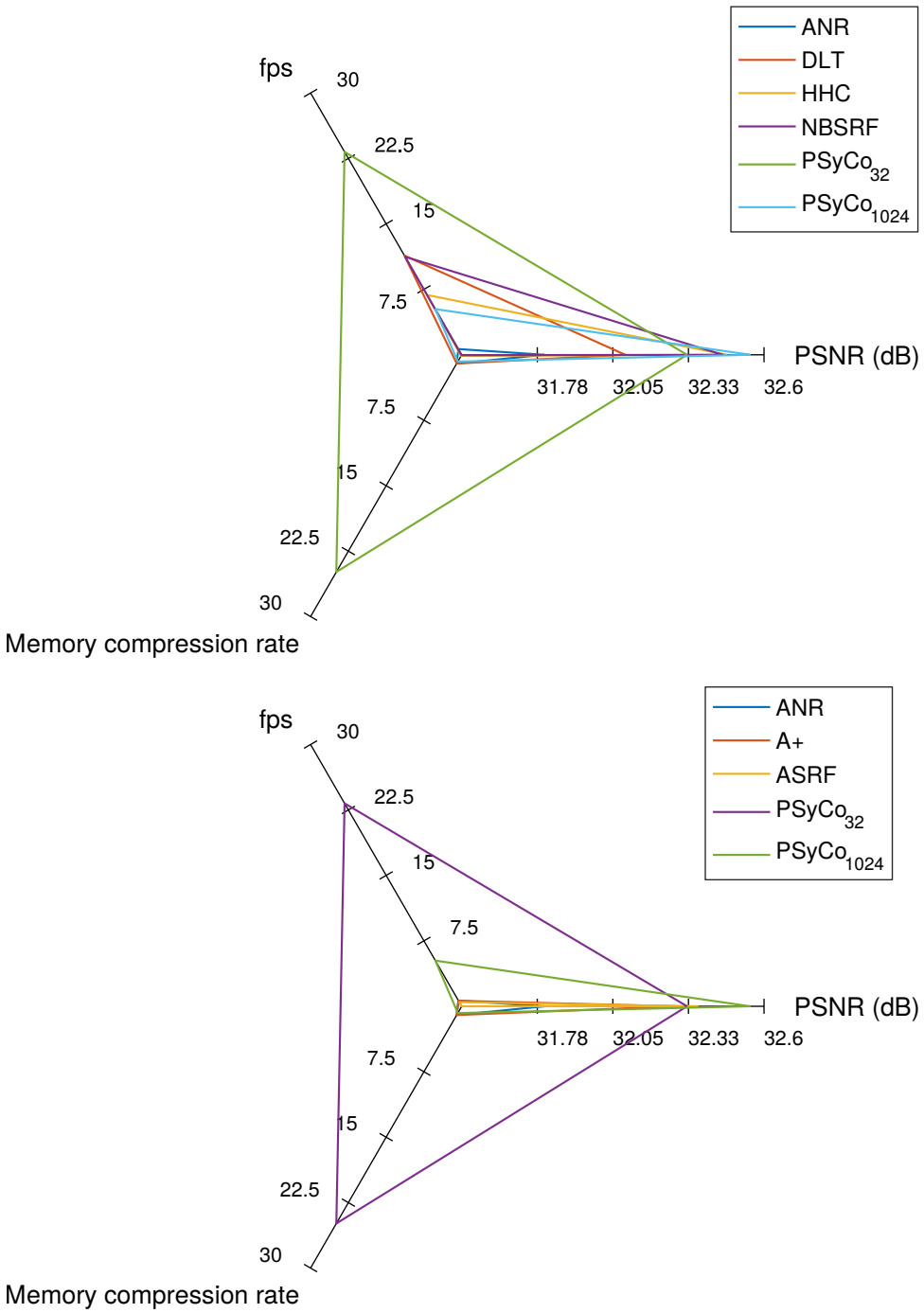


Figure 9.21: Plot of the Memory Compression Rate, frames per second (fps) and PSNR. **Top**: Evolution of the algorithms developed within this dissertation. **Bottom**: Comparison with other state of the art methods based on regression. Experiment run for a $\times 2$ upscaling factor in Set14.

Chapter 10

Conclusions

In this thesis we have addressed the problem of recovering a high-resolution image from its degraded and downscaled observation. This is an ill-posed problem which requires further constraints or prior knowledge in order to be tractable. In this thesis we tackle it by *learning* the low-to-high resolution correspondence from natural image statistics in the form of low-resolution and high-resolution patch examples.

Our contributions build on top of the coupled dictionary approach to SR of Yang et al. [95] and the Anchored Neighborhood Regression of Timofte et al. [74]. The contributions of this thesis can be summarized as follows:

Naive Bayes Adaptive Dictionaries

We present a Naive Bayes formulation to construct adaptive dictionaries with atoms that are semantically related to the content of the input image. We use training sub-image regions in order to preserve texture consistency, and we characterize each of them by densely extracting SIFT features. During inference, we use the efficient local Naive Bayes Nearest Neighbor approach to avoid estimating the probability density functions over all our training regions, and instead only do so for those regions in the local neighborhood of the input descriptors. The dictionaries obtained with our approach outperform those build by randomly sampling patches, even though this comes at the computational cost of training a dictionary per image (Chapter 4). Results of this contribution have been published in the British Machine Vision Conference (2013) [60]. We revisit our local NBNN with more emphasis on computational efficiency in our NBSRF.

Dense Local Training

We present a novel training scheme for regression-based SR. This new training approach is based on dense, fully collaborative neighbor embedding, which fits better the ℓ_2-regularizer present in ANR [74] and other linear regression methods. We propose to use the sparse dictionary atoms as anchor points to the manifold, but form the neighborhoods with raw manifold samples from a more extensive training pool. By doing so, we find tighter neighborhoods which, consequently, fulfill better the

local condition necessary for locally linear embedding. Additionally, a higher number of local independent measurements is available and we can control the size of the neighborhoods, i.e. it is not upper-bounded by the dictionary size. This contribution is key to obtain better fitted regression functions that yield remarkable quality improvements at no computational cost. Results of this contribution have been published in the Asian Conference on Computer Vision (2014) [62].

Spherical Hashing

We proposed a novel search strategy based on the data-dependant Spherical Hashing algorithm [31]. We train a set of hashing functions on patch statistics so that the manifold is partitioned in a balanced way. We then label the anchor points (and their associated regression functions) with hash codes so that, during inference, only a reduced set of hashing functions need to be computed for each input patch in order to find a regressor. This approach resulted in substantial speed-ups with respect to exhaustive search strategies. Results of this contribution have been published in the Asian Conference on Computer Vision (2014) [62].

Half-Hypersphere Confinement

We further analyzed the features and the metrics involved during the regression process. We studied the importance of antipodal invariance in our search space, and recommended the use of the cosine similarity over Euclidean distances. In order to be able to use any fast search structure, we propose a novel transform which boosts the antipodal invariance in the Euclidean space. This enables the Spherical Hashing to be antipodally invariant. The regressors obtained with cosine similarity show a neat gain in PSNR over those obtained with Euclidean distance. Furthermore, our antipodally invariant Spherical Hashing is optimally adapted to the regressor search as the drop in quality when compared to an exhaustive search is residual. Results of this contribution have been published in the Transactions on Image Processing [53] and in the Winter Conference on Applications of Computer Vision (2016) [54].

Naive Bayes SR Forest

We present a novel method for example-based SR, based on hierarchical manifold learning. We design a regression forest based on bimodal trees, where antipodal patches are effectively clustered together and both children subnodes have comparable homogeneity, thus leading to an overall better space sampling. In order to further extend the accuracy of the local linearizations of the coarse-to-fine mapping, we propose to use tree ensembles and select the optimal regression tree based on a Local Naive Bayes criterion. Results of this contribution have been published in the International Conference on Computer Vision (2015) [65].

Dihedral Symmetry Collapse

We present a new method for regression-based SR that builds around a novel manifold collapsing transform κ. This transform eliminates the undesired variability of the manifold due to the dihedral group of symmetries (i.e. rotation, vertical and horizontal reflections) and the antipodal symmetry. The dihedral group is specially suitable for SR as it is scale invariant and easily invertible. We perform a frequency analysis of the dihedral group in the DCT domain, where the group members are mapped as a combination of transpose and sign changes. Through a modification of the Symmetric Transform of Zabrodsky et al. [96] we collapse the 16 variations induced from the dihedral group and its antipodal extensions into a single primitive. The complexity of our proposed κ is inherently low, as it requires as little as 3 inner products and a matrix re-ordering. We exhaustively test our transform and also compare it with other recent state of the art methods. We consistently obtain $\times 16 - \times 32$ smaller dictionaries when aiming at a certain PSNR. For a fixed dictionary size, we greatly improve in terms of quality both objectively and qualitatively. Our method with 1024 atoms greatly surpasses the state of the art in terms of PSNR and IFC, and with a 32 atoms dictionary (i.e. reduced model size) we achieve competitive quality while being an order of magnitude faster. Results of this contribution have been published in the Computer Vision and Pattern Recognition Conference (2016) [55].

10.1 Future Work

PSyCo, our latest contribution, presents a unified framework for the incorporation of manifold structure knowledge, i.e. antipodal and dihedral symmetries, is highly flexible and it has great potential to be further refined. In this section we propose several ideas that could inspire some future work.

Gradient features in the transformed domain

We used patches without mean as input features for the regression in our latest work. The use of 1-st and 2-nd order gradient features as the ones in [97, 76, 74, 54, 62, 53] together with our PSyCo is not straightforward, as the derivatives of the dihedral group elements need to be accounted. Including gradient features and eventually a PCA compression scheme can improve performance and reduce memory usage, specially for high magnification factors.

Cascaded regression

Cascading stages of piece-wise linear regression has been applied to other problems with success [16], and has been proposed recently for SR [33, 75]. Extending our

PSyCo SR by adding extra cascaded regression stages is likely to improve the resulting quality, even though this will come at the cost of higher execution time and a bigger model size.

PSyCo and Spherical Hashing

A direct approach to speed-up PSyCo SR is creating a fast search structure that adapts well to the manifold distribution, in the same way we did with our proposed antipodally invariant Spherical Hashing or the bimodal tree, but in this case with support for dihedral invariance as well. Including the PSyCo transform within the Euclidean distance calculations of Spherical Hashing could be a good starting point in that direction.

Texture synthesis

The recent work of [64] opens an interesting discussion with respecth the minimization over the squared error or any other pixel-wise fidelity term common in many SR approaches. In their work, they propose a combination of automated texture synthesis with a perceptual loss focusing on creating realistic textures rather than optimizing for a pixel-accurate reproduction. The usage of such strategy can be beneficial to SR imaging, specially to deal with high-frequency stochastic textures which are unlikely to be recovered by any conventional SR method. A combination of both approaches through a deeper understanding of the scene (e.g. discerning textures from edges or flat areas) might bring together the best of the two worlds.

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311–4322, November 2006.

[2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 372–379 vol.2, 2000.

[3] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, September 2002.

[4] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24:1–24:11, July 2009.

[5] S. Bernard, L. Heutte, and S. Adam. On the selection of decision trees in random forests. In *Proc. Int. Joint Conf. Neural Networks*, pages 302–307, June 2009.

[6] M. Bevilacqua, A. Roumy, C. Guillemot, and M. Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proc. of the British Machine Vision Conf.*, pages 135.1–135.10. BMVA Press, 2012.

[7] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[8] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proc. IEEE Int. Conf. Computer Vision*, pages 1–8, October 2007.

[9] I. Bosch, J. Salvador, E. Pérez-Pellitero, and J. Ruiz-Hidalgo. An epipolar-constrained prior for efficient search in multi-view scenarios. In *Proc. European Signal Processing Conf.*, 2014.

[10] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[11] H. Chang, . Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, page I, June 2004.

[12] D. Dai, Y. Wang, Y. Chen, and L. Van Gool. Is image super-resolution helpful for other vision tasks? In *Proc. IEEE Winter Conf. Applications of Computer Vision*, pages 1–9, March 2016.

[13] V. De Silva and J. B Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Technical report, Stanford University, 2004.

[14] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[15] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[16] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1078–1085, June 2010.

[17] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision – ECCV 2014*, volume 8692 of *Lectures Notes in Computer Science*. Springer, January 2014.

[18] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(2):295–307, February 2016.

[19] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6):797–829, 2006.

[20] K. Engan, S. O. Aase, and J. H. Husøy. Multi-frame compression: Theory and design. *Signal Process.*, 80(10):2121–2140, October 2000.

[21] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[22] R. Fisher. Dispersion on a sphere. *Proc. of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 217(1130):295–305, 1953.

[23] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Trans. Graph.*, 30(2):12:1–12:11, April 2011.

[24] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, March 2002.

[25] William T. Freeman, Egon C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Int. Journal of Computer Vision*, 40(1):25, October 2000.

[26] Y. Freund, S. Dasgupta, M. Kabra, and N. Verma. Learning the structure of manifolds using random projections. In *Advances in Neural Information Processing Systems*, pages 473–480, 2007.

[27] X. Gao, K. Zhang, D. Tao, and X. Li. Image super-resolution with sparse neighbor embedding. *IEEE Trans. on Image Processing*, 21(7):3194–3205, July 2012.

[28] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Proc. IEEE Int. Conf. Computer Vision*, pages 349–356, September 2009.

[29] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 225–232, June 2011.

[30] K. He and J. Sun. Computing nearest-neighbor fields via propagation-assisted kd-trees. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 111–118, June 2012.

[31] J. P. Heo, Y. Lee, J. He, S. F. Chang, and S. E. Yoon. Spherical hashing. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2957–2964, June 2012.

[32] K. Hornik, I. Feinerer, M. Kober, and C. Buchta. Spherical k-means clustering. *Journal of Statistical Software*, 50(10):1–22, 2012.

[33] Y. Hu, N. Wang, D. Tao, X. Gao, and X. Li. Serf: A simple, effective, robust, and fast image super-resolver from cascaded linear regression. *IEEE Trans. on Image Processing*, 25(9):4091–4102, September 2016.

[34] B. Huang, W. Wang, M. Bates, and X. Zhuang. Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy. *Science*, 319(5864):810–813, 2008.

[35] J. B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 5197–5206, June 2015.

[36] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM.

[37] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graph. Models Image Process.*, 53(3):231–239, April 1991.

[38] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 29(6):1153–1160, December 1981.

[39] P. Kontschieder, S. R. Bulò, M. Pelillo, and H. Bischof. Structured labels in random forests for semantic labelling and object detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(10):2104–2116, October 2014.

[40] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Comput.*, 15(2):349–396, February 2003.

[41] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Proc. Int. Conf. on Neural Information Processing Systems*, pages 801–808, Cambridge, MA, USA, 2006. MIT Press.

[42] Z. Lin and H. Shum. Fundamental limits of reconstruction-based superresolution algorithms under local translation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):83–97, January 2004.

[43] Z. Lin and H. Shum. Response to the comments on "fundamental limits of reconstruction-based superresolution algorithms under local translation". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(5):847–, May 2006.

[44] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Information Theory*, 28(2):129–137, March 1982.

[45] X. Lu, H. Yuan, P. Yan, Y. Yuan, and X. Li. Geometry constrained sparse coding for single image super-resolution. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1648–1655, June 2012.

[46] R. J. Marks II. *Handbook of Fourier analysis & its applications*. Oxford University Press, 2009.

[47] S. McCann and D. G. Lowe. Local naive Bayes nearest neighbor for image classification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3650–3656, June 2012.

[48] J. McNames. A fast nearest-neighbor algorithm based on a principal axis search tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(9):964–976, September 2001.

[49] J. F. Murray and K. Kreutz-Delgado. Learning sparse overcomplete codes for images. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 46(1):1, January 2007.

[50] V. P. Namboodiri, V. D. Smet, and L. V. Gool. Systematic evaluation of super-resolution using classification. In *Proc. Visual Communications and Image Processing*, pages 1–4, November 2011.

[51] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 – 3325, 1997.

[52] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[53] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Antipodally invariant metrics for fast regression-based super-resolution. *IEEE Trans. Image Processing*, 25(6):2456–2468, 2016.

[54] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Half hypersphere confinement for piecewise linear regression. In *Proc. IEEE Winter Conf. on Applications of Computer Vision*, 2016.

[55] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. PSyCo: Manifold span reduction for super resolution. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016.

[56] G. Peyré. Manifold models for signals and images. *Comput. Vision Image Understanding*, 113(2):249–260, February 2009.

[57] E. Phan-huy Hao. Quadratically constrained quadratic programming: Some applications and a method for solution. *Zeitschrift für Operations Research*, 26(1):105–119, 1982.

[58] J. Platt. Fastmap, metricmap, and landmark mds are all nystrom algorithms. In *AISTATS*, 2005.

[59] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon. Metric regression forests for human pose estimation. In *Proc. of the British Machine Vision Conf.* BMVA Press, 2013.

[60] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Bayesian region selection for adaptive dictionary-based super-resolution. In *Proc. British Machine Vision Conf.*, 2013.

[61] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Accelerating super-resolution for 4k upscaling. In *IEEE Proc. Int. Conf. Consumer Electronics*, 2015.

[62] E. Pérez-Pellitero, J. Salvador, I. Torres-Xirau, J. Ruiz-Hidalgo, and B. Rosenhahn. Fast super-resolution via dense local training and inverse regressor search. In *Computer Vision – ACCV 2014*, volume 9005 of *Lectures Notes in Computer Science*. Springer, January 2014.

[63] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 40(8):1–15, 2008.

[64] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *IEEE Proc. Int. Conf. Computer Vision*, 2017.

[65] J. Salvador and E. Pérez-Pellitero. Naive Bayes Super-Resolution Forest. In *IEEE Proc. Int. Conf. Computer Vision*, 2015.

[66] J. Salvador, E. Pérez-Pellitero, and A. Kochale. Robust Single-Image Super-Resolution using Cross-Scale Self-Similarity. In *Proc. IEEE Int. Conf. Image Processing*, 2014.

[67] J. Salvador, E. Pérez-Pellitero, and A. Kochale. Fast single-image super-resolution with filter selection. In *Proc. IEEE Int. Conf. Image Processing*, 2013.

[68] S. Schulter, C. Leistner, and H. Bischof. Fast and accurate image upscaling with super-resolution forests. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3791–3799, June 2015.

[69] O. Shahar, A. Faktor, and M. Irani. Space-time super-resolution from a single video. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3353–3360, June 2011.

[70] H. R. Sheikh, A. C. Bovik, and G. de Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Trans. on Image Processing*, 14(12):2117–2128, December 2005.

[71] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1874–1883, June 2016.

[72] M. Shimano, T. Okabe, I. Sato, and Y. Sato. *Video Temporal Super-resolution Based on Self-similarity*, pages 411–430. Springer London, London, 2013.

[73] A. Srivastava, A.B. Lee, E.P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, 2003.

[74] R. Timofte, V. De, and L. V. Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proc. IEEE Int. Conf. Computer Vision*, pages 1920–1927, December 2013.

[75] R. Timofte, R. Rothe, and L. V. Gool. Seven ways to improve example-based single image super resolution. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1865–1873, June 2016.

[76] R. Timofte, V. De Smet, and L. V. Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Computer Vision – ACCV 2014*, volume 9006 of *Lectures Notes in Computer Science*. Springer, January 2014.

[77] I. Torres, J. Salvador, and E. Pérez-Pellitero. Fast approximate nearest-neighbor field by cascaded spherical hashing. In *Computer Vision – ACCV 2014*, volume 9006 of *Lecture Notes on Computer Science*. 2014.

[78] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. on Information Theory*, 53(12):4655–4666, December 2007.

[79] R. Tsai and T. Huang. Multi-frame image restoration and registration. *Advances in Computer Vision and Image Processing*, 1(2):317–3391, 1984.

[80] K. Turkowski. *Filters for Common Resampling Tasks*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[81] E. Van Reeth, I. WK Tham, C. Heng Tan, and C. L. Poh. Super-resolution in magnetic resonance imaging: A review. *Concepts in Magnetic Resonance Part A*, 40(6):306–325, 2012.

[82] M. J. Wainwright, E. P. Simoncelli, and A. S. Willsky. Random cascades on wavelet trees and their use in analyzing and modeling natural images. *Applied and Computational Harmonic Analysis*, 11(1):89 – 123, 2001.

[83] J. Wang, S. Kumar, and S. F. Chang. Semi-supervised hashing for scalable image retrieval. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3424–3431, June 2010.

[84] L. Wang and J. Feng. Comments on "fundamental limits of reconstruction-based superresolution algorithms under local translation". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(5):846, May 2006.

[85] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, January 2009.

[86] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing*, 13(4):600–612, April 2004.

[87] G. I. Webb. *Naïve Bayes*, pages 713–714. Springer US, Boston, MA, 2010.

[88] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, pages 1753–1760. Curran Associates, Inc., 2009.

[89] S. Winkler and P. Mohandas. The evolution of video quality measurement: From PSNR to hybrid metrics. *IEEE Trans. on Broadcasting*, 54(3):660–668, September 2008.

[90] A. Y. Yang, S. Rao, K. Huang, W. Hong, and Y. Ma. Geometric segmentation of perspective images based on symmetry groups. In *Proc. IEEE Int. Conf. Computer Vision*, pages 1251–1258 vol.2, October 2003.

[91] C. Yang, C. Ma, and M. Yang. Single-image super-resolution: A benchmark. In *Computer Vision – ECCV 2014*, volume 8692 of *Lectures Notes in Computer Science*. Springer, January 2014.

[92] C. Y. Yang and M. H. Yang. Fast direct super-resolution by simple functions. In *Proc. IEEE Int. Conf. Computer Vision*, pages 561–568, December 2013.

[93] J. Yang, Z. Lin, and S. Cohen. Fast image super-resolution based on in-place example regression. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1059–1066, June 2013.

[94] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[95] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. on Image Processing*, 19(11):2861–2873, November 2010.

[96] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(12):1154–1166, December 1995.

[97] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. *Curves and Surfaces*, January 2012.

[98] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. Journal of Computer Vision*, 73(2):213–238, 2007.

[99] K. Zhang, X. Gao, D. Tao, and X. Li. Multi-scale dictionary for single image super-resolution. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1114–1121, June 2012.

[100] L. Zhu, J. Shen, and L. Xie. Unsupervised visual hashing with semantic assistant for content-based image retrieval. *IEEE Trans. on Knowledge and Data Engineering*, PP(99):1, 2016.

Eduardo Pérez Pellitero

Computer Vision Researcher

Hafengasse 5
72070 Tübingen, Germany
+49 1744 527 677
eperez@tue.mpg.de
perezpellitero.github.com
perezpellitero
Born in Barcelona, 1989.

Experience

2012–2016 **Research Engineer**, *Technicolor R&I*, Hannover.
Enrolled in the Resolution Enhancement Group as a computer vision and image processing researcher.
Daily responsabilities included: Finding innovative solutions for problems in Technicolor's media production business, supporting Technicolor's academic research and filing related invention disclosures to strengthen the patent portfolio.
- 14 invention disclousures
- 11 peer reviewed publications (e.g. CVPR, ICCV, TIP)

Education

2012–2017 **Ph.D.**, *Leibniz Universität Hannover* and *Technicolor*, *defense in February*.
Manifold learning for Super-Resolution Upscaling.

2010–2012 **M.Sc. in Telecommunication Engineering & Management**, *Universitat Politècnica de Catalunya*.
Master thesis stay at KU Leuven: *Object detection using the chains model* supervised by L. Van Gool and R. Benenson.

2007–2010 **B.Sc. in Sound and Image Engineering**, *Universitat Politècnica de Catalunya*, obtained the *Third Best Academic Record Award*.

Computer skills

- Matlab and MEX interface
- C++ and OpenMP
- OpenCL
- GNU/Linux

Research interests

- Unsupervised learning
- Sublinear search (e.g. forest, hashing)
- Manifold learning
- Super Resolution, Inverse problems

Languages

Spanish, Catalan	Native	
English	Proficient	*C1 level in CEFR*
German	Lower intermediate	*B1 level in CEFR*

Scholarships and awards

2010 Third Best Academic Record Award.

2010, 2011 University Mentor Scholarship to best academic records.

2013 ICIP nomination to Outstanding Paper Award.

Publications

[1] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. PSyCo: Manifold span reduction for super resolution. In *CVPR*, 2016.

[2] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Antipodally invariant metrics for fast regression-based super-resolution. *IEEE Trans. Image Processing*, 25(6):2456–2468, 2016.

[3] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Half hypersphere confinement for piecewise linear regression. In *WACV*, 2016.

[4] J. Salvador and E. Pérez-Pellitero. Naive Bayes Super-Resolution Forest. In *ICCV*, 2015.

[5] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Accelerating super-resolution for 4k upscaling. In *ICCE*, 2015.

[6] E. Pérez-Pellitero, J. Salvador, I. Torres, Javier Ruiz-Hidalgo, and Bodo Rosenhahn. Fast super-resolution via dense local training and inverse regressor search. In *ACCV*, 2014.

[7] I. Torres, J. Salvador, and E. Pérez-Pellitero. Fast approximate nearest-neighbor field by cascaded spherical hashing. In *ACCV*, 2014.

[8] J. Salvador, E. Pérez-Pellitero, and A. Kochale. Robust Single-Image Super-Resolution using Cross-Scale Self-Similarity. In *ICIP*, 2014.

[9] I. Bosch, J. Salvador, E. Pérez-Pellitero, and J. Ruiz-Hidalgo. An epipolar-constrained prior for efficient search in multi-view scenarios. In *EUSIPCO*, 2014.

[10] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Bayesian region selection for adaptive dictionary-based super-resolution. In *BMVC*, 2013.

[11] J. Salvador, E. Pérez-Pellitero, and A. Kochale. Fast single-image super-resolution with filter selection. In *ICIP*, 2013.

References

Dr. Jordi Salvador
Technicolor scientific supervisor
✉ salvadormarcos@gmail.com
+49 1577 6871 748

Prof. Bodo Rosenhahn
Doctoral supervisor
✉ rosenhahn@tnt.uni-hannover.de
☎ +49 5117 625 316

Prof. Javier Ruiz-Hidalgo
Doctoral co-supervisor
✉ j.ruiz@upc.edu
☎ +34 9340 157 65

Axel Kochale
Technicolor Team Leader
✉ axel.kochale@t-online.de

Miscellaneous

2012–present **Fencing teacher**, *Arts of Mars*, Hannover.
Teaching traditional fencing style *Verdadera Destreza* (spanish school). Published a reference book:

[12] E. Pérez-Pellitero. *Iniciación a la Verdadera Destreza*. Lulu (self-published), 2012.

Music Jazz and classical studies on electric bass, piano and electric guitar.

Scuba Diving Open Water diver since 2014.