Algorithms for image saliency via sparse representation and multi-scale inputs image retargeting

Hoang, Minh Chau

2011

Hoang, M. C. (2011). Algorithms for image saliency via sparse representation and multi-scale inputs image retargeting. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50583>

<https://doi.org/10.32657/10356/50583>

Algorithms for image saliency via sparse representation and multi-scale inputs image retargeting

HOANG MINH CHAU

A thesis submitted to the Nanyang Technological University in
fulfilment of the requirement for the degree of Master of
Engineering

NANYANG TECHNOLOGICAL UNIVERSITY

2011

Abstract

Saliency detection is an important yet challenging task in computer vision. In this report we investigate the use of sparse coding over redundant dictionary for saliency detection. We attempt to present a small fraction of the growing knowledge regarding sparse representation over redundant dictionary and discuss some potential usage of this powerful tool for saliency detection task. We propose a new algorithm for saliency detection based on the likelihood that an image patch can be encoded sparsely using a dictionary learned from other patches. experimental results based on saliency ground of truth of 1000 real images shows a superior performance of the new algorithm in comparison with other existing saliency algorithms.

We also propose an image retargeting algorithm which is capable of combining the strength of the Shift-map framework and warping-based algorithms. The Shift-map algorithm experiences problems with extreme resizing ratio: important objects might be removed due to limited space in the output. We tackle this problem by introducing a stack of multi-scale inputs. This kind of input allows the Shift-map framework to produce output with great flexibility: regions can be removed or scaled in order to achieve the optimal and desired retargeted image. Experiments are conducted based on a benchmark image database to demonstrate potential power of this approach.

Acknowledgements

The special thank also goes to my supervisor, **Dr Deepu Rajan**. The supervision and support that he provided truly help my research and gave me much inspiration. My grateful thanks also go to our former research fellow, **Dr Hu Yiqun**. Discussion with him inspired me with many ideas and gave me many useful research experience. I also want to express my thankfulness to all our team members, who assisted me in many ways and provided help whenever necessary.

Last but not least, I would like to thank my girl friend, Anh Ngoc who has been supporting me throughout the project.

Contents

1	Introduction	7
1.1	Research Objectives	9
1.2	Contributions	10
1.3	Organization of the thesis	11
2	Saliency via sparse representation	13
2.1	Introduction	13
2.2	Review of saliency detection algorithms using sparse representation	14
2.2.1	Incremental Coding Length approach	15
2.2.2	Short-term representation saliency	17

2.2.3	Incremental Sparse Saliency approach	18
2.3	Review of the theory of sparse representation	20
2.3.1	Introduction	20
2.3.2	L1-minimization	22
2.3.3	Sparse representation via greedy algorithms	24
2.3.4	Solving (P_0)	25
2.3.5	Learning the dictionary	29
2.3.6	Sparse land model	31
2.4	Proposed saliency detection algorithm	33
2.4.1	Short-term K-SVD saliency	33
2.4.2	Sparse likelihood saliency	34
2.4.3	Experimental results	45
2.5	Conclusion	52
3	Image retargeting via multi-scale inputs	54
3.1	Introduction	54
3.2	Review of Shift-map retargeting	56

3.2.1	The frame-work	56
3.2.2	Graph-cut constraints	57
3.3	Multi-scale Shift-map for retargeting	60
3.3.1	The algorithm frame-work	61
3.3.2	Distortion map	63
3.3.3	Data constraints	64
3.3.4	Smoothness constraints	67
3.4	Experimental results and discussion	69
3.5	Conclusion	72
4	Conclusions and future work	74
4.1	Conclusions	74
4.2	Future work	76

List of Figures

2.1	Comparison between sparse coding via L1 norm and L2 norm	23
2.2	Similar patches	36
2.3	Solution of l1	40
2.4	Comparison of coefficients for natural image signals learned by K-SVD and ICA	41
2.5	Comparison of coefficients for natural image signals learned by K-SVD and ICA	46
2.6	Saliency map comparisons of various methods for 2 images. . .	47
2.7	Saliency map comparisons of various methods	48
2.8	ROC curve of different saliency algorithms	49
2.9	Comparison of performance of various algorithms	51

2.10	Some examples of saliency maps generated. From left to right: input image, ground-truth saliency map, ISS method [26], our method, ICL method [18], SICA method [34]	52
3.1	Output of Seam Carving retargeting algorithm.	55
3.2	Shift-map basic idea	57
3.3	Smoothness cost neighboring compariso	59
3.4	Stack of scale image sources	62
3.5	Distortion map patch samples	64
3.6	Retargeted output for battleship image of various retargeting methods	70
3.7	Retargeted output for pigeons image of various retargeting methods	71
3.8	Retargeted output images when resized to different size. . . .	71

Chapter 1

Introduction

Human visual system is known to have the ability to identify interesting regions in the a scene. Such regions are considered salient regions. Saliency detection is useful for computer vision applications such as object recognition [25] or image retargeting [16]. While humans have the ability to identify the saliency regions very well, it remains a very challenging problem for computer. One approach to tackle this problem is to make use of sparse representation of the input signals based on some dictionary [18], [34], [26], in which methods such as Independent Component Analysis (ICA) [4] and L1-minimization [35] have been used extensively. Coincidentally, theoretical studies [37] have suggested that V1 primary visual cortex can be efficiently represented by a sparse code based on an over-complete dictionary (or code-book) that resembles neurons found in area V1. Moreover, a redundant

dictionary which resembles a larger number of bases is preferred. However, finding a sparse representation using a redundant or over-complete dictionary is much more difficult and was considered intractable, hence the use of ICA. Fortunately, recent advances in compress sensing have provided more powerful tools to compute the sparse representation over redundant dictionary. These include Efficient Sparse Coding [24], Homotopy [9] and Orthogonal Matching Pursuit (OMP) [28]. These tools are very robust, efficient and have enabled many successful works in a wide range of applications, especially in computer vision. For instance, many tasks which were considered very difficult have been solved with state-of-the-art results, including face recognition [40], image denoising [11] and image super resolution [42]. In this thesis, we will investigate the use of sparse coding based on redundant dictionary to tackle the saliency detection problem.

One application of saliency detection is image retargeting in which a saliency map plays a very important role. Recently, with the development of a wide range of devices with different display sizes, image retargeting has become a very crucial task. The main challenge is to adapt an image intelligently into different size for optimal viewing experience in different devices. In only a few recent years, many interesting algorithms have been proposed to tackle this task, for example, Seam Carving [3], Shift-map Editing [29] or Mesh Parametrization [16]. There are 2 main approaches: Seam Carving (SC) related or warping-based approaches. SC related algorithms normally try to

remove/add pixels seamlessly to achieve the best result in the retargeted output. On the other hand, warping-based methods achieve the goal by resizing different regions in the image adaptively. While SC related algorithms have the power of removing unwanted objects easily, they lack the ability to resize an unimportant region if necessary. Consequently, for small retargeted output size, warping-based methods often deliver better results. In this thesis, we also attempt to develop a retargeting algorithm which has the advantages of both approaches.

1.1 Research Objectives

In this thesis, we would like to investigate the use of sparse coding based on redundant dictionary for the saliency detection application. The focus of the thesis is to determine what kind of dictionary can be used to represent the input sparsely and how the sparse representation can be used to determine the input image saliency.

Sparse representation has been used by Hou and Zhang in [18], Kong et al in [34]. However in such algorithms, ICA is used to learn the sparse representation, and in fact often gives dense codes due to a inverse matrix approach. Furthermore, the dictionary is often a square matrix, and hence can not be redundant. In order to obtain a sparse representation over redundant dictionary one will need specially designed algorithms such as Matching Pursuit

(MP) [43] or Least Square Regression (LAR or LARS) [10]. Sparse representation obtained by these algorithms are often sparser in comparison to one obtained by ICA.

The second objective is to develop an image retargeting algorithm that can inherit the power of the shift-map algorithm and warping-based methods. A truly flexible frame-work like Shift-map editing has many useful applications such as image inpainting or image completion, and is not limited to image retargeting. However, based on adding/removing pixels approach, scaling is not allowed, hence it often fails when the retargeted size is small. On the other hand, to resize the input image, warping based methods can scale each region in the input image adaptively. These methods are often limited to only resizing application.

1.2 Contributions

This thesis provides evidences that sparse representation over redundant dictionary may improve the saliency algorithm. More specifically, we show that by replacing the dictionary learned by ICA approach [4] by a dictionary learned by K-SVD approach [2], we have a saliency algorithm which outperforms the original Short-term saliency algorithm [34]. Furthermore, based on the observation that natural image patches are often highly correlated, we propose a new saliency algorithm which makes use of statistical perspective

of a L1-minimization with non-negative constraints. The key idea is that it is hard to represent a salient patch sparsely using other patches, while it is easy to do so with patches which are redundant in the image. experimental results show that our algorithm outperforms other state-of-the-art algorithms.

To introduce the warping effect to the Shift-map framework, we propose the use of multi-scale inputs. The power of Shift-map framework comes from the ability to shift pixels from the input to the retargeted output smartly, and hence the output can be considered as a re-organization of the input pixels. With the multi-scale inputs, the input source for the algorithm is no longer limited to the only original input image, but rather a stack of inputs with varying scales. Experimental results show that this approach can adapt the input image smartly by scaling or removing unimportant regions if necessary.

1.3 Organization of the thesis

Chapter 2 of the report reviews the basic theory of sparse coding and some of its applications with the focus of saliency detection. This chapter also provides brief descriptions of saliency detection algorithms which use sparse representation: Incremental Coding Length [18], Short-term representation saliency [34] and Incremental Sparse Saliency [26]. We suggest that using sparse representation will improve the quality of the saliency map in comparison to representation learned by conventional method like ICA. We also

exploit the statistical perspective of sparse representation to design a new saliency detection based on sparse likelihood measure. Experimental results are presented to evaluate the new approach, showing very promising potential power of sparse coding for saliency detection.

Chapter 3 will discuss the Shift-map framework for image retargeting and propose a new algorithm which extends the power of the algorithm by introducing multi-scale inputs. Experimental results show some samples of the new retargeting algorithm with interesting effect.

Finally, conclusions and directions for future work are presented in Chapter 4.

Chapter 2

Saliency via sparse representation

2.1 Introduction

The concept of a sparse representation or sparse coding is defined loosely in literature. Generally, a solution $\mathbf{s} \in \mathbb{R}^n$ of the linear system $\mathbf{Ds} = \mathbf{x}$ can be considered a sparse vector when it has only k nonzero indexes, k is small in comparison to n and $\mathbf{s}$ is often referred as a k -sparse vector. Such sparse solution can be obtained by using a redundant dictionary and seeking for the sparsest solution possible. Using sparse representation for saliency detection is an interesting approach, since such representation resembles the neuron in

the V1 cortex [37]. In comparison with a dense code, a sparse code has more discriminative power since information is concentrated in only a few bases.

In this chapter, we review some saliency detection algorithms which make use of sparse representation. We also review basic ideas and theory of sparse representation on redundant dictionary. Finally we propose new saliency detection methods which uses sparse representation on a redundant codebook learned from the input image.

2.2 Review of saliency detection algorithms using sparse representation

Several related algorithms which made use of sparse representation are discussed, including Incremental Coding Length (ICL) [18], Incremental Sparse Saliency (ISS) [26] and Short-term Sparse Representation Saliency (SSRS) [34]. It is noted that the term 'sparse representation' or 'sparse coding' is used here in a broad sense to be aligned with the literature. In some algorithms such as ICL or SSRS, the sparse representation is obtained via a matrix inverse approach using a given dictionary which is learned using standard methods like ICA. This is not to be confused with the sparse representation obtained from a line of algorithms from other approaches such as Matching Pursuit (MP) [43], Orthogonal Matching Pursuit (OMP) [28],

Least Square Regression (LAR or LARS) [10], Basis Pursuit (BP) [8] or Efficient Sparse Coding (ESC) [24]. While ICA often works with a full-rank and square dictionary, algorithms such as OMP or LARS are designed to work with a redundant dictionary. The problem of interest of these algorithms is not only to find a sparse representation, but rather the *sparsest* representation. As shown in literature as well as in our experimental results later, the coefficient learned by these algorithms is much sparser than the coefficient learned by ICA and hence lead to an improvement in saliency detection. To avoid confusion, methods which are used to obtain the sparse representation of each algorithm will be noted clearly for clarification and comparison purposes.

2.2.1 Incremental Coding Length approach

Hou and Zhang [18] learned a set of basis functions via ICA that gives sparse representation for each input image patch. Each basis of the learned dictionary is used as a feature in the saliency analysis. More specifically, the coefficient corresponding to an input image patch is determined by $\mathbf{s} = \mathbf{D}^{-1}\mathbf{x}$, where $\mathbf{s}$ is the coefficient and $\mathbf{x}$ is the image patch stacked as a vector and $\mathbf{D}$ is the learned dictionary. In the cortex representation, each non-zero coefficient corresponds to an activated neuron and how much energy that neuron consumes. Although a straight forward summing of all response of each feature will give a simple measure of energy consumed when an input patch is

introduced, this measurement is not very meaningful. Hou and Zhang proposes that we can redistribute the energy of each feature so that the input can be encoded more efficiently. The average response to all input patches can be considered as a probability function $\mathbf{P}$ in $\mathbb{R}^N$, N is the number of features, in which each index denotes the probability a feature i is excited. Incremental Coding Length (ICL) for a feature i is then defined as [18]

$$\mathbf{ICL}(p_i) = -H(\mathbf{P}) - p_i - \log p_i - p_i \log p_i, \quad (2.1)$$

where $H(\mathbf{P})$ is the entropy of $\mathbf{P}$, calculated as: $H(\mathbf{P}) = -\sum_{i=1}^N p_i \log(p_i)$, where p_i is the probability of feature i and N is the number of feature. Basically, the index i of $\mathbf{ICL}$ measure how much the entropy $H(\mathbf{P})$ changes if a new excitation is introduced to feature i . Intuitively, the more entropy is gained when a feature is activated, the more salient that feature is. Energy is then redistributed so that more energy is given to more salient feature. The saliency is then defined as

$$sal(i) = \sum_{j=1}^N d_j r_j, \quad (2.2)$$

where N is the number of feature, d_j is the ICL of feature j , r_j is the response of feature j to patch i . The saliency of a patch according to this equation is not constant but may vary over time depending on the input.

2.2.2 Short-term representation saliency

In literature, sparse representation is often obtained using a dictionary which is normally pre-defined e.g., Fourier or Wavelet bases, or trained using thousands of natural image patches. The dictionary is 'global' and is not changed nor adapted depending on the input. Kong et al [34] proposed that learning an adaptive dictionary based on the input image would provide a representation with better accuracy and hence improved the saliency detection quality. This type of representation is referred to as short-term representation since it is derived from information received in short period of time. Given patches sampled in overlapping manner from the input image, they trained a dictionary which can represent each patch sparsely using ICA method. Next the background firing rate (BFR) is defined as

$$BFR_j = \sum_{i=1}^N F_{ij}/M, \quad (2.3)$$

where F_{ij} the response of the j^{th} feature to the patch i^{th} and M is the number of input patches. Stacking BFR_j values together as a vector gives the average response of each atom in the dictionary. It is noted that this function is similar to one defined by Hou in [18] as the probability function of the feature activities. The feature activation rate (FAR) or the amount of energy consumed when a new visual input appears is defined as

$$FAR_i = \sum_{j=1}^N |F_{ij} - BFR_j|, \quad (2.4)$$

where N is the number of features. This short-term energy is used as the saliency value for the target patch. We refer to this algorithm as short-term ICA (or SICA).

2.2.3 Incremental Sparse Saliency approach

Inspired by saliency detection approaches that use the phenomenon of centre-surround contrast [20], Li et al [26] proposed a difference to surrounding scheme using sparse representation. The idea of the center-surround contrast is simple: the sparse representation of a patch using surrounding patches as the dictionary gives a cue of how different the patch is from its surrounding. To determine whether a patch is salient, the algorithm runs in 2 steps. First, patches in the surrounding area is densely sampled to form a redundant dictionary $\mathbf{D}$. Second, the center patch's representation using the dictionary is computed via L1-minimization method, i.e., finding $\mathbf{s}$ such that $\mathbf{Ds} = \mathbf{x}$, where $\mathbf{x}$ is the center patch vector. If the center patch is similar to its surrounding, the coefficient obtained will be very sparse. On the contrary, if the center patch is different from the surrounding patches, many atoms in the dictionary are required to approximate the center patch, resulting in a non-sparse coefficient. Denoting the patch of interest as p , the saliency $S(p)$

is then defined as the number of nonzero values of $\mathbf{s}$, i.e., the coding length of $\mathbf{s}$:

$$S(p) = \|\mathbf{s}\|_0, \quad (2.5)$$

where $\|\cdot\|_0$ is the l_0 -norm which counts the number of nonzero indexes and $\mathbf{s}$ is the sparsest coefficient that can be used to represent the center patch in terms of the surrounding patches. The coefficient $\mathbf{s}$ is obtained by solving the Lasso problem [35], which is the minimization problem of the sum of squared errors, i.e., $\|\mathbf{D}\mathbf{s} - \mathbf{x}\|^2$, with the bound on the sum of absolute values of the coefficients, i.e., $\|\mathbf{s}\|_1$. The problem has been found to be closely related to the problem of the finding sparsest coefficient where the bound is placed on $\|\mathbf{s}\|_0$ instead. Since this approach solves the Lasso problem to find $\mathbf{s}$, the coefficient found is sparser in comparison with coefficients used in other methods like ICL or SSRS. However, using the coding length for saliency to determine the difference between the center patch and the surrounding may not be entirely correct. The length of the sparse code, i.e., how many atoms of the dictionary are used to approximate the center patch in fact depends on the dimension of the subspace that the patch belongs to. For instance, it is possible to represent the center patch of an uniform blue region by only 1 single surrounding patch, while a patch belonging to a tree or cloudy region may lie in a higher dimensional subspace, hence more atoms will be needed. While both patches should be non-salient, the varying

number of dimension of their subspaces makes the salient measure unstable. Furthermore, in order to reach a stable solution, Lasso needs a dictionary which is sufficiently incoherent, a condition is often violated when sampled surrounding patches are highly similar and correlated (will be discussed in more details in section 2.3).

2.3 Review of the theory of sparse representation

2.3.1 Introduction

Given an input signal $\mathbf{x} \in \mathbb{R}^N$, one would like to find a linear expression of $\mathbf{x}$ in term of some dictionary (or codebook) $\mathbf{D} \in \mathbb{R}^{N \times K}$, i.e. solving $\mathbf{x} = \mathbf{D}\mathbf{s}$. Generally a sparse coefficient $\mathbf{s}$ is preferred, where the term sparse representation here is used loosely to describe a coefficient with only few nonzero indexes. One reason for this preference is that sparsity provides more information in comparison with the case where the nonzero coefficient is widespread across all bases. When the coefficient is dense, each basis carries only a little information about $\mathbf{x}$. On the contrary, when the coefficient is sparse, the information about $\mathbf{x}$ is more concentrated. This kind of representation is very informative, especially in object classification.

A popular approach to learn sparse representation is ICA, where independent components learned are statistical independent and non-Gaussian, providing sparse representation of the data. While the dictionary learned via ICA is often square and invertible, here we would like to focus on a case where the dictionary is highly redundant or over-complete i.e. $K \gg N$. Since the linear system $\mathbf{x} = \mathbf{D}\mathbf{s}$ is underdetermined, there are infinite solutions for $\mathbf{s}$. To look for meaningful information, one should find a sparsest solution which uses as few atoms as possible. To put formally, finding the sparsest solution for the redundant system requirement turns the original problem to the following NP-hard problem:

$$(P_0): \min_{\mathbf{s}} \|\mathbf{s}\|_0 \quad \text{subject to} \quad \mathbf{x} = \mathbf{D}\mathbf{s}, \quad (2.6)$$

where $\|\cdot\|_0$ is the l_0 -norm, which counts the number of nonzero indexes of $\mathbf{s}$. Researchers have been addressing this problem using 2 main approaches: using greedy algorithms or using a relaxation technique which replaces the l_0 -norm by a l_1 -norm to turn (P_0) to a tractable problem. This is a very interesting and fast moving topic, however to discuss in details all important discoveries in sparse representation is out of the scope of this thesis. In the following sections, we will only attempt to summarize some important results which are related to computer vision applications, especially saliency detection.

2.3.2 L1-minimization

The basic idea of solving (P_0) using a L1-minimization approach is to simply replace the l_0 -norm with the l_1 -norm and hence turns the (P_0) into the following (P_1) problem:

$$(P_1) : \min_{\mathbf{s}} \|\mathbf{s}\|_1 \quad \text{subject to} \quad \mathbf{x} = \mathbf{D}\mathbf{s}. \quad (2.7)$$

The relaxed problem can be solved via the following convex problem:

$$(P_1^\lambda) : \hat{\mathbf{s}} = \operatorname{argmin}_{\mathbf{s}} \frac{1}{2} \|\mathbf{D}\mathbf{s} - \mathbf{x}\|^2 + \lambda \|\mathbf{s}\|_1, \quad (2.8)$$

with a proper choice of λ , where λ is the parameter that controls the trade-off reconstruction error $\|\mathbf{D}\mathbf{s} - \mathbf{x}\|^2$ and the sparsity $\|\mathbf{s}\|_1$. The convex problem (P_1^λ) can be solved efficiently using linear programming tools. In a sense, the (P_1) problem can be viewed as an intermediate problem between the (P_0) problem and the well-known (P_2) problem which can be stated as

$$(P_2) : \min_{\mathbf{s}} \|\mathbf{s}\|_2 \quad \text{subject to} \quad \mathbf{x} = \mathbf{D}\mathbf{s}. \quad (2.9)$$

(P_2) minimizes the l_2 -norm instead of the l_1 -norm and gives us the familiar least-square solution: $\mathbf{s} = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{x}$.

Figure 2.1: Left: l_1 -minimization approach. Right: l_2 -minimization approach. s_0 is the desired sparsest solution. Figure is adapted from [40].

Geometrically speaking, minimizing a l_1 -norm will give a sparser solution than l_2 -norm minimization. Figure 2.1 demonstrates the geometry of l_1 -minimization in comparison with l_2 -minimization. Minimizing via the l_2 norm is equivalent to finding the solution by extending the l_2 ball until it touches the solution space. Hence, the result of this approach is not sparse, unless the solution space is perpendicular to the axes. On the other hand, the level sets of the l_1 norm is octahedral, aligned with the coordinates axes. Extending the l_1 ball until it touches the solution space naturally has a sparser result than l_2 minimization approach.

In general, algorithms which follow this direction can solve the problem (P_1) exactly and efficiently, especially for large-scale linear systems [9], [24], [22]. The question remained is of course, whether the solutions of (P_1) and (P_0) coincide.

2.3.3 Sparse representation via greedy algorithms

The basic purpose of greedy algorithms like MP [43], OMP [28], LARS [10] is to find the sparsest solution $\mathbf{s}$ of the linear system: $\mathbf{Ds} = \mathbf{x}$, given the target $\mathbf{x}$ and the dictionary $\mathbf{D}$. The idea behind MP is as simple as its name suggests: for each iteration, the algorithm finds an atom in the dictionary which is the most correlated with the residual, which is initialized as the target $\mathbf{x}$. After each iteration, a new atom of the dictionary is added to the active set $\mathbf{S}_k$ with the coefficient which has the value of $\langle \mathbf{r}^k, \mathbf{d}_i \rangle$, where $\mathbf{r}^k = \mathbf{x} - \mathbf{Ds}^k$ is the current residual and $\mathbf{s}^k$ and $\mathbf{d}_i$ are the coefficient and the most correlated dictionary atom at the iteration k , respectively.

While MP is often slow, a simple modification gives us the OMP algorithm: OMP would update the coefficient such that the residual after each iteration should be uncorrelated (i.e. orthogonal) instead. Some other approaches have been proposed alternatively, such as LARS or Homotopy. In fact these algorithms are also heuristic based approach and can be seen as a modification of MP or OMP mentioned above. For instance, the difference between LARS and OMP is that LARS would demand that the correlation is constant instead:

$$|\langle \mathbf{r}^k, \mathbf{d}_i \rangle| = \text{const}, \forall i \in \mathbf{S}_k. \quad (2.10)$$

Homotopy (also known as LARS-LASSO) [36] would demand that a new index can enter or old index can leave the active set and for each $i \in \mathbf{S}_k$ the following is maintained:

$$|\langle \mathbf{r}^k, \mathbf{d}_i \rangle| = \text{const} > \max_{j \notin \mathbf{S}_k} |\langle \mathbf{r}^k, \mathbf{d}_j \rangle|. \quad (2.11)$$

It is easy to see that, indeed Homotopy is nothing more than a variation of the original MP algorithm. While originally Homotopy is used for overdetermined case, Tsaig [36] has shown that Homotopy can be used in underdetermined setting and in fact has a nice k -step stopping property.

2.3.4 Solving (P_0)

In both L1-minimization and greedy approaches, the question remained is whether these proposed algorithms can deliver the correct (i.e. sparsest) solution that we want. Greedy approaches such as MP or OMP are of course heuristic and hence the solution reached may be suboptimal. Similarly, the biggest doubt for L1-minimization is whether solving (P_1) gives us the same solution of (P_0) .

Surprisingly, a series of work by Bruckstein, Donoho, Tsaig [6], [7], [36] shows that in fact under some settings it is possible to recover the correct solution of (P_0) exactly, given the dictionary is sufficiently incoherent and the solution is

sufficiently sparse. In order to illustrate the idea of "sufficiently sparse" and "sufficiently incoherent" condition, we will briefly describe some observations by Donoho et al [6].

Define the *spark* of a matrix as the smallest number of columns in the matrix which are linearly dependent. It can be shown that if $\|\mathbf{s}\|_0 < \text{spark}(\mathbf{D})/2$, then $\mathbf{s}$ is the sparsest solution possible. To see this, let $\mathbf{s}'$ be another solution which satisfies $\mathbf{D}\mathbf{s}' = \mathbf{x}$, we have $\mathbf{D}(\mathbf{s}' - \mathbf{s}) = 0$ or $\mathbf{s}' - \mathbf{s}$ is in the null-space of $\mathbf{D}$. By the definition of *spark* we have

$$\|\mathbf{s}'\|_0 + \|\mathbf{s}\|_0 \geq \|\mathbf{s}' - \mathbf{s}\|_0 \geq \text{spark}(\mathbf{D}), \quad (2.12)$$

or the sum of the number of nonzero indexes of $\mathbf{s}'$ and $\mathbf{s}$ is greater than the *spark* of $\mathbf{D}$. This means if $\|\mathbf{s}\|_0$ is smaller than $\text{spark}(\mathbf{D})/2$ then $\|\mathbf{s}'\|_0$ has to be greater than $\text{spark}(\mathbf{D})$ and then in turn greater than $\|\mathbf{s}\|_0$ i.e., $\mathbf{s}$ is the sparsest possible solution. It is easy to see a direct implication is that, let $\mathbf{D} \in \mathbb{R}^{n \times M}$ be the collection of M vectors in general positions, $M > n$, any solution $\mathbf{s}$ with $\|\mathbf{s}\|_0 < n/2$ or fewer can be considered the uniquely sparsest solution. The term 'general positions' here means these vectors do not satisfy any special linear relations or fall into any degenerate structure, and hence the minimum number of linearly dependent vectors is n .

However, in practice it is not easy to evaluate the *spark* of a matrix. Donoho introduced the mutual coherence of a matrix $\mu(\mathbf{D})$ as the maximum correla-

tion between any two normalized columns and it can be proved that [6]

$$\text{spark}(\mathbf{D}) \geq 1 + \frac{1}{\mu(\mathbf{D})}. \quad (2.13)$$

Obviously, if we have a solution $\mathbf{s}$ such that $\|\mathbf{s}\|_0 < \frac{1}{2}(1 + \mu(\mathbf{D})) \leq \frac{1}{2}\text{spark}(\mathbf{D})$ then this solution is the sparsest possible. The mutual coherence is easy to compute and hence provides a practical way to verify the correctness of the solution found. When the matrix is highly incoherent i.e. $\mu(\mathbf{D})$ is small, the bound for $\text{spark}(\mathbf{D})$ is then higher, increasing the size of the set of uniquely sparse solutions.

In fact, the above relationship between the dictionary incoherence, sparseness of the solution $\mathbf{s}$ is much more important. Let $\mathbf{D}\mathbf{s} = \mathbf{x}$ and $\mathbf{D}$, $\mathbf{s}$ satisfy the requirement i.e. $\mathbf{D}$ must be sufficiently incoherent and $\mathbf{s}$ is sparse enough. To put formally

$$\|\mathbf{s}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right), \quad (2.14)$$

then $\mathbf{s}$ is in fact guaranteed to be recovered using sparse coding algorithms. For instance Bruckstein [6] proves that if the dictionary is incoherent and the solution is sufficiently sparse i.e. equation (2.14) holds, algorithms from both previously mentioned approaches (specifically, OMP and BP) can find it exactly. It can even be shown that the solution can be found after only

$k = \|\mathbf{s}\|_0$ steps using Homotopy algorithm [41]. Indeed there are evidences that two seemingly different approaches are in fact closely connected.

The mutual coherence can be seen as a property that shows how much the linear system behaves differently from the orthogonal system. When $\mu(\mathbf{D})$ approaches 0, $\mathbf{D}$ in fact describes a orthogonal system. This is aligned with the work from Candes [7], which shows that if $\mathbf{D}$ satisfies a Restricted Isometry Property (RIP) with a constant δ_K , the difference between the solution obtained by solving (P_1) and the true solution is very small or for very sparse solution it completely vanishes. The RIP condition basically requires the system to behave approximately like an orthogonal system [7] i.e.,

$$(1 - \delta_K)\|\mathbf{s}\|_2^2 \leq \|\mathbf{D}\mathbf{s}\|_2^2 \leq (1 + \delta_K)\|\mathbf{s}\|_2^2, \quad (2.15)$$

holds for all K -sparse signals $\mathbf{s}$. It is easy to see that if δ_K is very small, $\mathbf{D}$ behaves approximately like an orthogonal system i.e. $\|\mathbf{s}\|_2^2 \approx \|\mathbf{D}\mathbf{s}\|_2^2$. Furthermore, it can be shown that under proper setting the solution can still be recovered exactly as long as the corrupted fraction is not too large, despite an arbitrary large-size error corruption. This important contribution is one key factor that leads to success in computer vision applications such as face classification [40].

2.3.5 Learning the dictionary

Choosing a proper dictionary is crucial in applications which use sparse representation, especially in computer vision. In most cases, the dictionary is not given beforehand but rather designed based on the input data. Researchers have been trying to obtain the dictionary by assembling input signals directly [40] or by training a large database of input samples [2]. Training a dictionary using a pool of input signals is hard yet an important problem, which arises naturally when the number of input signals is large and one need a concise dictionary to describe them sparsely. Generally, finding both the dictionary and the sparse representation at the same time given the input signals is formulated as

$$\underset{\mathbf{D}, \mathbf{S}}{\operatorname{argmin}} \sum_i^N \|\mathbf{s}_i\|_0 \quad \text{subject to} \quad \mathbf{D}\mathbf{S} = \mathbf{X}, \quad (2.16)$$

where $\mathbf{S}$ is formed by concatenating all coefficients $\mathbf{s}_i$ and $\mathbf{X}$ is formed by concatenating all N input signals $\mathbf{x}_i$. The minimization with respect to both $\mathbf{D}$ and $\mathbf{X}$ is hard and hence the problem is often simplified into 2 solvable minimization problems with respect to $\mathbf{D}$ or $\mathbf{X}$ only. A popular approach to tackle this problem is then to alternate between 2 simpler steps: fixing the dictionary $\mathbf{D}$ and find the sparsest coefficients for all inputs, fixing the coefficients $\mathbf{S}$ and then find the dictionary $\mathbf{D}$. When the dictionary $\mathbf{D}$ is fixed, the problem can be decoupled to N problems which minimize only one signal

at a time and can be effectively solved by any algorithm mentioned previously. The second step is to update the dictionary given the coefficients and the input signals. K-SVD algorithm [2] solves this by an SVD approach, result in an algorithm which resembles the K-mean clustering. Efficient sparse coding algorithm [24] solves this step with an Lagrange dual approach, providing an fast and efficient algorithm to learn the dictionary.

Here we will briefly describe the K-SVD approach: in the first step, the dictionary is fixed and K-SVD uses OMP (or any pursuit algorithm) to find the coefficient. In the second step, K-SVD updates 1 atom of the dictionary at a time. Let $\mathbf{s}_k$ denotes the row vector of $\mathbf{S}$ corresponding to the dictionary atom $\mathbf{d}_k$, this atom can be updated by looking at minimizing the objective function

$$\|\mathbf{E}_k^R - \mathbf{d}_k \mathbf{s}_k^k\|_F^2, \quad (2.17)$$

where $\mathbf{s}_R^k$ denotes the result discarding all zero indexes of $\mathbf{s}_k$, $\mathbf{E}_k^R$ is the error obtained when using only atom $\mathbf{d}_k$ to approximate the input. The minimization can be done directly via SVD where the atom $\mathbf{d}_k$ is updated in such a way that the updated coefficient $\hat{\mathbf{s}}_R^k$ is forced to have the same support as the original $\mathbf{s}_R^k$. Hence, it is worth noted that in K-SVD, the second step the coefficient $\mathbf{S}$ is not strictly preserved. This is to compare with algorithm like ESC [24] where the dictionary is updated via a Lagrange dual approach and

the coefficient does not change at this step.

2.3.6 Sparse land model

In order to apply sparse coding to computer vision applications, one will often need a model to describe the image signals. One popular model can be described as following: given input signal $\mathbf{x} \in \mathbb{R}^N$ and assume that it can be expressed by a over-complete dictionary $\mathbf{D} \in \mathbb{R}^{N \times K}$, $K > N$ i.e. $\mathbf{Ds} = \mathbf{x}$ where the coefficient $\mathbf{s} \in \mathbb{R}^K$ has no more than k_0 nonzero indexes. As mentioned previously, to find an exact solution $\mathbf{s}$ given $\mathbf{D}$ and $\mathbf{x}$ is difficult, and it is common that some relaxation of the sparseness constraint and error constraint is applied. For instance, the condition $\mathbf{Ds} = \mathbf{x}$ can be relaxed to an approximation $\mathbf{Ds} \approx \mathbf{x}$, and the error is allowed to be up to ϵ . Similarly, the solution need not to be the sparsest, but only sparse to some extent. To be specific, one may characterize the model by setting a parameter k_0 as the upper limit for the sparseness of $\mathbf{s}$. To put formally:

$$\text{Find } \mathbf{s} \text{ Subject to } \|\mathbf{Ds} - \mathbf{x}\|^2 < \epsilon, \quad \|\mathbf{s}\|_0 < k_0. \quad (2.18)$$

This model can be regarded as $\mathcal{M}(\mathbf{D}, k_0, \epsilon)$, and is referred as *Sparse Land* model and is widely used in computer vision [27] [2]. Hence the basic idea of the model is that any image can be expressed as a linear combination of only a few atoms of an over-complete dictionary. The approximation parameter

ϵ allows the flexibility of model, and is useful in case of noisy input. This model can be used in many image processing tasks such as compression or denoising. For instance, in compression problem any signal $\mathbf{x}$ can be stored using only k_0 numbers after solving $\mathbf{x}$ using $\mathcal{M}(\mathbf{D}, k_0, \epsilon)$ [2]. In this case, the model has a reconstruction error of up to ϵ . By relaxing ϵ and restricting k_0 further more, we have a compression scheme with larger error and higher compression rate.

The problem described in equation (2.18) is not a hard problem and can be solved via many sparse coding algorithms mentioned previously. For instance, the error of approximation ϵ can be the input to the OMP algorithm as the parameter for the stopping rule. Similarly, a small modification of the BP algorithm which relaxes the constraint $\mathbf{Ds} = \mathbf{x}$ to:

$$\min_{\mathbf{s}} \|\mathbf{s}\|_1 \text{ subject to } \|\mathbf{Ds} - \mathbf{x}\|_2 \leq \epsilon, \quad (2.19)$$

giving us the well-known Basis Pursuit Denoising (BPDN) algorithm [8].

2.4 Proposed saliency detection algorithm

2.4.1 Short-term K-SVD saliency

According to [37], V1 primary visual cortex can be modelled using a sparse coding, which can be dense codes or local codes, where neurons give very selective response to the input. If the neurons are represented by dense codes, an input will excite many neurons and each neuron carries little information. Therefore local codes are preferred. However, local codes require a large number of neurons and hence are computationally intractable. In compress sensing terms, large number of neurons is translated to a highly redundant dictionary $M \times N$, where N is much greater than M . A local neuron response, can then be seen as a sparse solution $\mathbf{S}$ of the equation $\mathbf{X} = \mathbf{D}\mathbf{S}$, where $\mathbf{X}$ is the input. Earlier this problem was regarded as intractable, and hence a sparse solution is normally found through standard methods such as ICA. As discussed in section 2.3.2, sparse representation in ICA approach is often obtained by calculating the inverse matrix of $\mathbf{D}$, i.e., $\mathbf{s} = \mathbf{D}^{-1}\mathbf{x}$ and hence is not sparse. Luckily, as discussed in section 2.3, recent advance in compress sensing has shown many promising efficient and robust algorithms to the problem. The sparse codes found by these algorithms seem to be a better model for the neural response. For instance, Lee [24] has proposed an algorithm to compute the sparse coding efficiently, and show some promising experiments to show similar behaviours of the sparse coding result and a V1

neural response.

Given patches sampled from the input, we would like to learn a dictionary which can represent each patch sparsely. We assume that a sparser representation is preferred. Hence instead of using standard method such as ICA, we propose to use recent advanced methods such as K-SVD [2] or efficient sparse coding [24]. To evaluate the power of the sparser representation, we use the same model as in [34] to determine the saliency. Let s_i be the coefficient corresponding to each patch after the training process, the saliency is simply defined as

$$sal(p_i) = \|s_i - mean(s)\|_1, \quad (2.20)$$

where p_i represents the target patch and $mean(s)$ is the average of all coefficients. Equation (2.20) can be seen as equivalent to equation (2.3) and (2.4).

2.4.2 Sparse likelihood saliency

The dictionary training process is to find a dictionary which can best sparsely represent all input signals. With the sparse constraint and a small size dictionary, signals are not approximated well equally. Signals which appear to be very redundant will be approximated better since the algorithm tries to reduce the approximate error as the whole. Furthermore, from a statistic point of view, we assume that saliency may be measured using the proba-

bility that a signal belongs to a sparse land model. A signal is considered salient if the probability that it can be represented by the model is low. In this section we will propose a new approach in which the image statistics is exploited via L1-minimization to determine the saliency of each image patch.

2.4.2.1 L1-approximation of natural image patches with non-negativity constraint

Let $\mathbf{S} = \{\mathbf{x}_i \in \mathbb{R}^n, i = 1 \dots N\}$ be a collection of natural image signals sampled from an input image simply by stacking pixels of image patches of size $\sqrt{n} \times \sqrt{n}$ in lexicographic manner. It has been known that natural image signals belonging to the same class exhibit a degenerate structure i.e. lie in or near a low-dimensional subspace [39]. Suppose the input signals are all normalized to unit l_2 -norm 1, we observe that a set of signals obtained from similar patches in this manner are highly correlated. For instance, similar patches shown in figure 2.2 despite having varying brightness and pattern, has an astonishingly high minimum dot product with their normalized mean: 0.9842 and 0.9945 respectively. This is to compare with observation done by [40] where face images of the same person have a minimum dot product of 0.723 with their normalized mean.

Suppose all signals collected from the input are stacked together to form a dictionary $\mathbf{D} \in \mathbb{R}^{n \times N}$. Assume $\mathbf{D}$ can be partitioned into M sub-matrix $\mathbf{D} = \{\mathbf{C}_i\}_{i=1}^M$, $\mathbf{C}_i \in \mathbb{R}^{n \times N_i}$, each contains N_i similar signals which exhibit

Figure 2.2: Left: original image. Top right: patches sampled from the sky region with minimum dot product of 0.9842 with their normalized mean. Down right: patches sampled from the grass region with minimum dot product of 0.9945 with their normalized mean. To illustrate how concentrated they are, the minimum dot product between all patches sampled from the image is 0.2354.

such degenerate structure. If $\mathbf{C}_i$ is sufficient and redundant enough, we may assume a new signal $\mathbf{y}$ drawn from $\mathbf{C}_i$ can be linearly represented by signals in $\mathbf{C}_i$ i.e. $\mathbf{x} = \mathbf{C}_i \mathbf{s}_i$ where $\mathbf{s}_i \in \mathbb{R}^{N_i}$. Hence, in term of the dictionary $\mathbf{D}$, $\mathbf{x}$ can be expressed as

$$\mathbf{x} = \mathbf{D} \mathbf{s}, \quad (2.21)$$

where $\mathbf{s} \in \mathbb{R}^N$, $\mathbf{s} = [0, \dots, \mathbf{s}_i^T, \dots, 0]^T$, that is all indices of $\mathbf{s}$ are zeros except those associated with $\mathbf{C}_i$. This kind of coefficient is often referred to as "block-sparse" representation [14] [13] [12] in literature.

In comparison to other approaches like SSRS or the algorithm described in

section 2.4.1, this approach does not have a dictionary training step which is very costly especially when algorithms like K-SVD are used. A dictionary assembled from all input patches is highly redundant and coherent hence violates most of the conditions which make sparse coding works (section 2.3). However in practice, it has been shown that indeed this approach can still give very high successful rate. For instance, one may recall that the dictionary used in the successful work by Wright et al [40] is created in similar manner i.e. concatenating input face images directly. If dictionary atoms are in general positions, any sparse representation has less than $n/2$ nonzero indexes can still be considered recoverable, despite the fact these atoms can form highly concentrated and correlated clusters.

It is also noted that although the dictionary as a whole is coherent, the dictionary atoms are not correlated uniformly (figure 2.2). While patches in each sub-matrix $\mathbf{C}_i$ can be highly correlated, we assume that patches from different sub-matrices are still sufficiently uncorrelated. Similar sparse representation settings have been studied in literature. For instance, Eldar [12] establishes that if patches are drawn from a union of subspaces which satisfy a similar incoherent condition mentioned in section 2.3, the sparse representation can be reliably recovered. Elhamifar et al [14] proves that if patches belong to independent subspaces, a sparse representation obtained via L1-minimization using a dictionary created by concatenating input samples is block-sparse exactly i.e. having only nonzero indexes in corresponding $\mathbf{C}_i$.

If the given subspaces are only known to be disjoint, one can still recover the block-sparse representation exactly if the principle angles between any 2 subspaces satisfies some certain bound [13]. However, such assumption is still too strong in our case. Hence based on the nature of the natural image patches, we further require that $\mathbf{s}$ is non-negative. The problem of interest hence turns to be

$$(P'_1) : \hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \quad \|\mathbf{D}\mathbf{s} - \mathbf{s}\|^2 + \lambda \|\mathbf{s}\|_1, \quad \mathbf{s} \succeq 0, \quad (2.22)$$

where $\mathbf{s} \succeq 0$ means all indexes of $\mathbf{s}$ are greater than 0. This is reasonable since $\mathbf{D}, \mathbf{x} \succeq 0$ and a contribution of *negative patch* to the target patch is hard to interpret. This assumption is also aligned with observation from Wright [40] that even without explicit constraint, the coefficient tends to be non-negative. In our experiments, we observe that such constraint indeed improves the sparse representation obtained. Furthermore, since nonzero indexes are preferred to be associated with similar patches, algorithms that encourage the bases to be orthogonal, like OMP, should not be used. Instead, one should use algorithms which encourage the atoms to be as much correlated to the target as possible, for example Homotopy [36]. In our experiment, we use L1-minimization algorithm provided by [23] for efficiency purpose.

Such sparse representation is very informative in comparison to represen-

tation learned by conventional methods such as Independence Component Analysis (ICA) which is used in some other saliency algorithms [18] [34]. In comparison, L1-minimization can achieve much higher sparsity, unlike ICA where the coefficient is often spread across all bases. For instance, one may expect non-zero indices in $\mathbf{s}$ should only correspond to patches which are most similar to $\mathbf{y}$. In fact, this simple constraint allows the recovered sparse representation to be more reliable and informative. For illustration, figure 2.3 shows an example where a target patch can be approximated by only a few patches which are similar to the input patch the most. The target patch is approximated via L1-minimization using a dictionary $\mathbf{D}$ formed by sampling patches from the image in overlapping manner. To avoid trivial solution a small area surrounding the target patch is excluded. The result of approximation is illustrated by black rectangles with varying transparency which indicate how much weight is given a patch in order to approximate the target patch.

2.4.2.2 Saliency measurement via statistical perspective

With the constraint of non-negative coefficient, it is easy to see that natural image signals are modeled in a way that signals from the same 'class' span a tight and highly concentrated convex cone. As discussed previously, such structure can be exploited by a L1-minimization using a dictionary $\mathbf{D}$ formed by all signals sampled from the input image. Since $\mathbf{D}$ contains all information

Figure 2.3: Top row: The target patch (blue rectangle) is approximated by similar patches (black rectangles) in the image. Bottom row: sparse coefficient learned by L1-minimization with non-negativity constraints, results obtained using algorithm from [22].

about the input $\mathbf{y}$, given a new input signal one may be interested to learn some statistical information of $\mathbf{y}$ given $\mathbf{D}$. For instance, if signals which are similar to $\mathbf{y}$ appear to be redundant in $\mathbf{D}$, it is likely that a very sparse approximation of $\mathbf{y}$ can be found. On the other hand, if $\mathbf{y}$ is rare and does not belong to any cone a sparse representation is very hard to achieve (figure 2.4). From the figure we can see that it is hard to get a sparse representation for signal like $\mathbf{y}$ which does not belong to neither $\mathbf{C1}$ or $\mathbf{C2}$. If $\mathbf{y}$ is forced to be approximated by $\mathbf{C1}$ or $\mathbf{C2}$, the representation will not be sparse.

Unfortunately we do not know any information given about how many partitions should be in the $\mathbf{D}$ nor how tight each cone should be. Hence we propose to use a statistical approach to measure how likely an input $\mathbf{x}$ can

Figure 2.4: Signals belong to the set spanned by 'bouquet' **C1** or **C2** are easier to approximate with a sparse representation.

be sparsely represented by **D**. It is known that minimizing the equation given in problem (P3) corresponds to a MAP inference in a probabilistic model with a Laplacian prior [15]. To see this, let **s** have a Laplacian distribution, i.e. $p(\mathbf{s}) = \frac{\lambda}{2} e^{-\lambda|\mathbf{s}|_1}$. Based on the Bayes rule we have $p(\mathbf{s}|\mathbf{D}, \mathbf{x}) \propto p(\mathbf{x}|\mathbf{D}, \mathbf{s})p(\mathbf{s})$. The MAP estimate of **s** is then:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \{-\log p(\mathbf{s}|\mathbf{x}, \mathbf{D})\} \quad (2.23)$$

$$= \underset{\mathbf{s}}{\operatorname{argmin}} \{-\log p(\mathbf{x}|\mathbf{s}, \mathbf{D}) - \log p(\mathbf{s}|\mathbf{D})\}. \quad (2.24)$$

Assume **s** is independent we have $-\log p(\mathbf{s}|\mathbf{D}) = \lambda|\mathbf{s}|_1 + c$ where c is some constant depends on the parameter λ . It is easy to see that with an appropriate Gaussian distribution model on $p(\mathbf{s}|\mathbf{x}, \mathbf{D})$ solving the problem in

equation (2.24) is equivalent to the L1-minimization in the form of equation (2.8). For instance, let $p(\mathbf{s}|\mathbf{x}, \mathbf{D}) = \frac{1}{2\sqrt{\pi}}e^{-\frac{1}{2}\|\mathbf{D}\mathbf{s}-\mathbf{x}\|^2}$, we have $-\log p(\mathbf{s}|\mathbf{x}, \mathbf{D}) = \frac{1}{2}\|\mathbf{D}\mathbf{s} - \mathbf{x}\|^2 + d$ where d is some constant. Equation (2.24) turns to be

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{D}\mathbf{s} - \mathbf{x}\|^2 + \lambda |\mathbf{s}|_1 + C \right\}, \quad (2.25)$$

where C is some constant. The error of approximation $\|\mathbf{D}\mathbf{s} - \mathbf{x}\|^2$ is then a good indicator of how likely $\mathbf{x}$ can be sparsely represented by $\mathbf{D}$.

Let $p(i) = \frac{1}{2\sqrt{\pi}}e^{-\frac{1}{2}\|\mathbf{D}_i\mathbf{s}_i-\mathbf{x}_i\|^2}$ be the likelihood measurement $p(\mathbf{x}_i|\mathbf{s}_i, \mathbf{D}_i)$ of the event patch i belongs to the sparse model with dictionary $\mathbf{D}_i$, the rarity/saliency of patch i can be measured by

$$\operatorname{sal}(i) = 1 - \bar{p}(i), \quad (2.26)$$

where $\bar{p}(i)$ is the probability $p(i)$ normalized to the range 0-1. In this case, $\mathbf{D}_i$ is a dictionary formed by all signals in $\mathbf{S}$ except for $\mathbf{x}_i$. It is noted that to satisfy the non-negativity constraint, $\mathbf{s}_i$ is the coefficient learned by solving problem (P'_1) .

2.4.2.3 Incorporating intensity information

Using a l^1 -minimization approach will require that all input signals have to be normalized to l^2 -norm unit length to avoid l^1 scale problem where a vector with shorter length is simply easier to approximate than a longer one. By doing so we lose the brightness information of each patch. Although some tolerance to the intensity is good, a very dark and very bright patches should not be treated similarly. The intensity information should be integrated in a way that the effect of patch brightness can be controlled and the convex cone model can still be maintained i.e., similar patches with similar intensity should still be highly correlated and concentrated to form a 'bouquet'.

Given that the intensity of a patch can vary from 0 to 1 after normalization, a natural way to achieve the requirement is to map these values to a set of polar vectors with radius 1 and angle vary from θ_{min} to θ_{max} . These vectors are of the same length and the intensity difference can be indicated by the angle between them i.e. large brightness difference corresponds to large angle or small inner product between two vectors. Let $\mathbf{x}_i$ be the original vector, a new vector $\mathbf{x} = [\mathbf{x}_i \mathbf{i}_i]$ can be formed by concatenating the original vector with the intensity vector $\mathbf{i}_i$. Since all intensity vectors has the same length, the final vector has the length of $\|\mathbf{x}_i\|^2 + \|\mathbf{i}_i\|^2$ i.e. the contribution of intensity to the length remain unchanged. The inner product between two similar vectors (before normalization) is then $\mathbf{x}_i^T \mathbf{x}_j + \mathbf{i}_i^T \mathbf{i}_j$. It is easy to see in this framework the 'bouquet' framework is not violated and the size of the cone

can be easily controlled by varying the range $[\theta_{min}, \theta_{max}]$.

2.4.2.4 Adaptive dictionary

One common problem with saliency algorithms is that it is often hard to identify large size object. Many algorithms based on center-surround contrast often highlights strong edges as salient and misses the interior of the object. The convex cone model we propose can handle this situation very easily. In case of large size object, a salient patch may have its surrounding similar to itself, but yet in term of a global context this patch is still very distinctive. On the other hand, the presence of similar patches in the dictionary results in a good approximation and hence low saliency value. Excluding the surrounding in the dictionary is not a good remedy either, since a patch which is very different from the surrounding is definitely salient. Therefore, one may want to remove only similar patches which lie in the surrounding area of the target patch. Based on the proposed model, the similarity can be indicated by simply computing the inner product between the center patch and its surrounding patches. Any surrounding patches with an inner product with the center patch higher than a value β should be eliminated from the dictionary.

2.4.3 Experimental results

The experiment is executed using a database of 1000 images with saliency ground-truth masked by human [1]. Several existing methods were also chosen for comparison purpose.

2.4.3.1 Sparseness of coefficients

First, we evaluated the sparseness property of 2 training algorithms ICA and K-SVD. We used K-SVD algorithm provided by Elad [11] and fastICA algorithm provided in [4]. Natural input signals were obtained by sampling patches from natural image in overlapped manner. Each patch of size $n \times n$ was then stacked lexicographically to form a vector $\in R^{n \times n}$. Treating these vectors as the training data, we obtained two dictionaries learned using K-SVD and ICA that could approximate all input signal sparsely. In both cases, the size of the dictionary for K-SVD is fixed at 192 (for direct comparison with SSRS later). Figure 2.5 shows typical coefficients for the same signal learned using K-SVD and ICA. K-SVD shows a very clear improvement over the sparseness of the coefficient. Interestingly, the same experiment using random input signals instead of natural patches did not show any significant difference between two learning methods. A possible explanation is that random input signals are independently generated and are wide-spread in the space. Hence it is hard to approximate all signals sparsely. On the

Figure 2.5: Comparison of coefficients learned by K-SVD and ICA training algorithms. Input signals are patches sampled from a natural image.

other hand, natural patches sampled from an image tend to cluster into subspaces, for instance patches sampled from a sky region belong to a low-rank subspace. They are highly correlated and although the number of input patches is large, it is possible to use a small dictionary to represent each patch sparsely. Efficient sparse coding algorithm by [24] also shown similar results.

2.4.3.2 Experimental results for short-term K-SVD saliency

We also compares saliency maps obtained using K-SVD approach some other saliency algorithms. Besides short-term ICA method, some well-known algorithms are also chosen including Itti's well-known algorithm [20], Spectral Residual [17], Frequency-Tuned [1], Superpixel Clustering and Saliency Propagation (SCSP) [30], Incremental Coding Length (ICL) [18] and Incremental

Figure 2.6: Saliency map comparisons of various methods for 2 images.

Sparse Saliency (ISS) [26]. Some sample images from the experimental result are shown in figure 2.6 and 2.7. It can be seen that Itti’s method is a center-surround contrast approach hence saliency is often drawn to the edge and high contrast region (figure 2.6(b)). SCSP depends heavily on the segmentation step and returns bad saliency map when incorrect segmentation is given. On the other hand, Frequency-Tuned algorithm is sensitive to color difference and assigns high saliency value for region with distinctive color, which is not always correct (figure 2.6 (a)). Spectral residual is prone to unique edge patterns and sometimes fails to identify the salient object (figure 2.7 and instead gives high saliency value to the object’s outer edge. ICL sometimes mistakes the background as saliency region when the background contains complicated patterns (figure 2.7). Overall, our method returns better results in comparison with ICA method, providing a sparser and more

Figure 2.7: Our method demonstrate a very good saliency map in comparison with other methods. In the input image the salient object is distinctive in both pattern and color, but the background is also complex.

Ours	ICA	Itti	SCSP	SR	FT	ISS	ICL
0.85161	0.82375	0.78377	0.9326	0.76698	0.83179	0.90167	0.8527

Table 2.1: Average area under the ROC curve of various methods

accurate saliency map. Objects with distinctive pattern and color are identified very well, especially objects with small to average size. This is because under sparse coding via K-SVD, patches from different objects are likely to be grouped into separate sub-spaces.

To evaluate overall performance, we used the Receiver Operating Characteristic (ROC) [5]. Figure 2.8 displays the ROC curves of saliency based

Figure 2.8: ROC curve for comparison of saliency detection between different algorithms. Our method performs better than all methods except SCSP that uses segmentation and ISS which make use of multi-scale image inputs.

K-SVD and ICA methods, obtained by averaging ROC result curves of 1000 images in the database. The average area under the ROC curve is shown in table 2.1. The figure clearly shows that saliency based K-SVD algorithm demonstrates a better ROC curve with larger area under the curve in comparison to SICA method. However in term of average area under the curve, our method is worse than ICL and SCSP method. By the nature of the patch-based approach, our method results in a fuzzy area around the edge of the object. SCSP contains a step for segmentation of the image and hence it may not be fair to compare our algorithm with it. The proposed algorithm does not contain any preprocessing like segmentation. However, we believe that a segmentation step will increase the accuracy of the saliency map drastically since the object order can be identified very well in most cases. It

is also notable that in case of "fuzzy" object (image in figure 2.6(a)) which causes trouble for segmentation algorithm, our method gives a better result. Besides, the short-term representation approach is essentially a difference to average approach, and hence suffers if the object size is large. This is also an issue for center-surround contrast approach like ICL.

2.4.3.3 Experimental results for sparse likelihood saliency

To verify the sparse likelihood saliency algorithm proposed in previous section, patches of size 8×8 are sampled from the input image with overlapping of 4 pixels. For each patch, a raw vector of size 196 is concatenated with a 2D vector carrying the average intensity of the patch to form our input signals set. For each signal, a dictionary is constructed by discarding the target signal. The dictionary is further improved by discarding similar signals in a surrounding area of 5 times the patch size, in which the parameter β is set to 0.7. For each pair of signal and dictionary, the problem (P1') in equation (2.22) is solved using algorithm provided by [22] with parameter λ is set to 0.05.

To evaluate the performance, we use the Receiver Operating Characteristic (ROC) method from [5]. Figure 2.9 shows that our algorithm outperforms all state-of-the-art algorithms, showing better consistency with the ground-truth. In terms of average area under the curve, our algorithm also yields the best result (table 2.2). It is noted that the performance of this algorithm is

Figure 2.9: Average ROC curves of all methods on 1000 images with human-masked ground-truth.

Ours	short-term ICA	Itti	ISS	ICL
0.9293	0.82375	0.78377	0.90167	0.8527

Table 2.2: Average area under the ROC curve of various methods

also significantly better than the algorithm described in section 2.4.1 which has only the score of 0.85161.

We show some samples of our saliency in comparison with saliency map of ISS (the next best algorithm in ROC evaluation). Unlike ISS, our saliency map is not attracted to strong edges. Due to the global excluding surrounding approach, our algorithm works best with salient object with relatively large size (figure 2.10).

Figure 2.10: Some examples of saliency maps generated. From left to right: input image, ground-truth saliency map, ISS method [26], our method, ICL method [18], SICA method [34]

2.5 Conclusion

In this chapter we have investigated the use of sparse representation over redundant dictionary for saliency detection application. A summary of recent saliency algorithms such as ICL[18], SICA [34] and ISS [26] was given. These are algorithms which use sparse representation in the approach. We also reviewed some basic ideas of sparse coding theory which is related to our topic. Experimental results were provided to show that new sparse coding approach resembles the V1 visual cortex better, with a sparser coefficient. A new algorithm which makes use of the sparse representation over redundant dictionary was discussed and experimental results was presented. The experiment demonstrated that the new algorithm shows a promising result with better performance in comparison with algorithms which use a similar

approach.

We also propose an algorithm which leverages the L1-minimization approach to learn the image statistics to measure the saliency value of each image patch. By assuming redundant image patches are more likely to have sparse representation based on a dictionary constructed from other patches in the image, we show how a L1-minimization-based framework can naturally lead to a robust algorithm which outperforms other existing methods. Although the framework is relatively simple and saliency calculation is straight-forward, it is very easy to extend and integrate new information to improve the result.

Chapter 3

Image retargeting via multi-scale inputs

3.1 Introduction

In the past few years there has been significant research in image retargeting application. The purpose of retargeting is to adapt the image so that it can be displayed in different devices with different screen sizes, mostly mobile devices with small screen size. Hence the image needs to be resized in such a way that important content is still preserved and displayed properly (figure 3.1). In only a few years, a wide range of approaches and ideas were proposed to tackle the problem, such as Seam Carving (SC) [3], Shift-map Editing (SM) [32],

Scale and Stretch (SNS) [38], Multi-Operator (MO) [33] to name a few. Most algorithms can be categorized to SC related or warping-based methods. SC related algorithms normally try to remove/add pixels seamlessly to achieve the best result in the retargeted output. On the other hand, warping-based methods achieve the goal by resizing different regions in the image adaptively. A common approach is that the salient object is preserved as much as possible and smooth or unimportant regions are resized to reach the target size. In the section below we discuss a method which is a hybrid algorithm between SC related algorithms and warping-based algorithms.

Figure 3.1: Output of Seam Carving retargeting algorithm, from right to left: original image (350 x 300), resized image (290 x 300), resized image (240 x 300).

This chapter of the report is organized as follows: Shift-map frame-work which is the basis of our algorithm is reviewed in Section 3.2, the Multi-scale SM is discussed in Section 3.3 and finally the experimental results are presented in Section 3.4.

3.2 Review of Shift-map retargeting

3.2.1 The frame-work

SM was introduced by Pritch et al [29] for image retargeting. It formulates the image retargeting problem as a multi-label graph-cut optimization. Each pixel in the output image is considered a 'node' in the graph, each node is connected to its 4 spatial neighbors. Each shift to a pixel in the input image is a 'label'. Hence, by labeling an output node with a label from the input, the algorithm essentially 'shifts' a pixel from input to the output. The output image is the result of choosing a proper collection of pixels from the input. In case of resizing to smaller image, some input pixels will be discarded and only a small subset of pixels are selected. Unlike many image resizing algorithms, the pixel arrangement nature of the algorithm allows direct extension to applications such as object removal, inpainting or object rearrangement.

Define $T(p, l) = t(x, y)$ as a shift operator which finds the source pixel of output pixel p when it is given the label l . Given an output pixel $p(x, y)$, a shift label $l(u, v)$, the source pixel in the input is calculated as

$$I(x', y') = T(p, l) = I(x + u, y + v). \quad (3.1)$$

In other words, the pixel in location $p(x, y)$ in the output image will take the

Figure 3.2: Two output pixels are given the same label L1 have the same spatial relationship in the input. The whole output region is given the same label L, hence it is shifted from the whole input region.

value of the pixel in the location $T(p, l)$ in the input image. It is noted that if 2 output pixels are given the same label $l(u, v)$, their spatial relationship is the same in the input. Hence if all pixels in a region of the output is assigned the same label, the effect is equivalent to selecting a region in the input to appear in the output (figure 3.2).

3.2.2 Graph-cut constraints

In Shift-map framework, the retargeted output is found by seeking an optimal labeling of the graph. This can be done by graph-cut data and smooth energy minimization. Denote a label mapping of an output pixel p by $M(p)$, the objective of a graph-cut algorithm is to minimize the total energy

$$E = \alpha \sum E_d(M(p)) + \sum E_s(M(p), M(q)), \quad (3.2)$$

where E_d is the data cost which constrains a specific labeling and E_s is the smoothness cost that controls the continuity of the labeling over two neighboring nodes p and q .

An example of a data term constraint is:

$$E_d(p, l) = S(T(p, l)), \quad (3.3)$$

where $S(t)$ is a saliency map which returns high value on pixels which are not important. This type of constraints prefer some labeling over others, and in this case is useful to make sure important object to appear in the output. Non-salient region or background have a high data cost and hence any shift to these regions is expensive. The data term is very useful in preventing certain shifts or prefer some pixels to appear in the output. Similarly other constraints can be associated with data term easily, for instance prohibiting shifting outside of the input or forcing the right-most and left-most columns of the output to come from the corresponding columns of the input. For object removal application, any shift to the removed object is set to infinity.

The smoothness cost controls the algorithm in another aspect. As mentioned above, when two output pixels are given the same label, the labeling is considered 'smooth' and no artifact appears since their values are taken from two neighboring pixels in the input. However when two pixels which are not neighbors in the input are located together as neighbors in the output,

Figure 3.3: Smoothness cost neighboring comparison. Left: original image. Right: output image. If 2 neighboring pixels in the output are assigned label L1 and L2, the smooth cost is computed by the difference between their corresponding neighbors in the original image.

artifacts may appear. The smoothness cost measures the difference between two corresponding spatial neighbors (figure 3.3). The smoothness cost E_{sm} can be defined as:

$$E_{sm}(T(p_1, l_1), T(p_2, l_2)) = R(I, T_1, T_2) + R(\nabla I, T_1, T_2), \quad (3.4)$$

where $R(I, T(p_1, l_1), T(p_2, l_2))$ is the neighboring difference, defined as:

$$R(I, T(p_1, l_1), T(p_2, l_2)) = [I(T(p_1, l_1)) - I(T(p_2, l_1))]^2 + [I(T(p_1, l_2)) - I(T(p_2, l_2))]^2. \quad (3.5)$$

To understand the equation, it is noted that $I(T(p_1, l_2))$ is simply the mapped pixel if the output pixel p_1 has the same label as p_2 . If the neighbor of 2 pixels are identical, the smoothness cost is 0.

Basically, graph-cut algorithm allows us to find an minimal energy among in all possible label allocations. The energy is computed over each label allocation of each output pixel and each pair of neighboring output pixels. Hence, energy minimization via graph-cut is a global optimization performed over individual pixels. Based on this frame-work, Shift-map Editing algorithm for video is also proposed [19]. A better salient object preservation Shift-map for image retargeting can also be found at [21].

3.3 Multi-scale Shift-map for retargeting

Most retargeting algorithms fall into two categories: warp or non-scaled image retargeting. Non-scaled image approach works by looking at the resized image as a collection of pixels from the input images, hence no scaling is possible. Representatives of this approach are Seam Carving [3], Shift-map Editing [29]. The performance of non-scaled method is often excellent when retargeted width is not very small in comparison with the input. This kind of algorithms often removes unimportant object or region in the image, in many cases unnoticeable. However when the retargeting output is small (especially smaller than the main object), artifacts are unavoidable since some parts of the salient object will have to be removed. To tackle this problem, a family of algorithms with scaling ability is introduced. The main idea of this approach is that unimportant region will be resized more, leaving im-

portant region resized less or none at all (ideal case). Although showing a clear success in small output size, scaling algorithms often distort the image and artifact is more visible. For this reason, we propose an algorithm which combines both properties, and present some nice useful applications of the ability to remove unimportant regions and scaling at the same time.

3.3.1 The algorithm frame-work

In order to introduce warping into the output, several image sources are used instead of one. Shifting from a region of a scaled image source will give the effect of compressing that region in the output. The output image can be thought of as a combination of several scaled input images (figure 3.4). A good algorithm should keep the important object in the original size as much as possible, and compress/remove other regions to compensate the small resizing size instead.

Let $\{r_i, i = 1 \dots n\}$ be the set of n ratios used to generate the source, a series of image sources $\{I_i, i = 1 \dots n, I_i = I \times r_i\}$ is generated as the input for the algorithm, where $I \times r_i$ is the result of resizing input I with a ratio r_i . Given a image source I_i with the size $w_i \times h_i$, let $L_i = \{l_i, i = 1 \dots N_i\}$ is the collection of labels that is assigned to this set of input pixels, where N_i is the total number of labels. Depending on the specific application of the algorithm, the number of labels may vary. When an output pixel is assigned

Figure 3.4: Input image is scaled to form a stack of image sources

a label that belongs to the collection L_i , a value of that pixel will be taken from the resized source i with ratio r_i . When a group of pixels belonging to a region in the output image is shifted using the same label l_i that belongs to L_i , we have the effect of warping that region to the scale r_i . For instance if all pixels in the output is given a label that shifts to a source i such that $I \times r_i = I_{output}$ then we have a simple uniform scaling of the input effect. Given a label $l(u, v)$ and a destination output pixel $p(x_o, y_o)$, we have a pixel source computed by:

$$O(x, y) = I_i(x_o + u, y_o + v), \quad (3.6)$$

where i the index of a image source in the stack is identified by which set of labels l belong to. The pixel mapping process is the same with Shift-map Editing algorithm, except for an extra needed step to identify which source the pixel is shifted from based on a given label.

3.3.2 Distortion map

A common issue in retargeting algorithm is to find a good balance between retaining the amount of information in the input and possible distortion. For instance by uniformly scaling the image, we may preserve all the information but the distortion is high. By seamlessly removing some pixels, the algorithm such as Seam Carving or Shift-map practically trade-off losing information to less distortion output. In warping based algorithms, some regions are 'compressed' to leave rooms for important objects, and they are ideally regions which shows little or no distortion after scaling. To incorporate such information to the algorithm, we describe a visual distortion measure which can determine how much a region is visually distorted after the image is re-sized. This visual distortion should be large for structure content and small for smooth or textured regions. The amount of distortion is computed by comparing corresponding patches in the scaled image with the original image (figure 3.5).

In order to measure the scaled distortion in a location in the resized image, the sum of squared difference (SSD) between patch sampled in this location and the corresponding patch in the original input is computed. To put formally:

$$d_p = SSD(I(p), I_r(p)), \quad (3.7)$$

where I_p and $I_r(p)$ are corresponding patches in the input and the resized

Figure 3.5: Sample patches at corresponding locations in original image and horizontal resized image

input respectively and r is the ratio that the image is resized. This measure decreases gradually as r approaches 1. It is easy to see that using this measure will give high value for region that possesses high contrast texture within itself.

3.3.3 Data constraints

Some basic requirements of the labeling in the Shift-map framework is naturally extended to Multi-scale Shift-map through the use of the data cost. The first data term is the simple out-of-bound constraint. This constraint is to make sure that a valid shift does not fall out of the boundary of the source images. The second data term is to make sure that the output boundary comes from the input source boundary. In this case we have multiple inputs, hence we require that the boundary of the output should come from the boundaries of one of the inputs. This requirement is not strictly applied, we

allow pixels at the boundary of the output to come from different sources' boundaries.

With multiple sources which represent the same input image in different scales, identifying which part in the stack is preferred to shift to is tricky. Based on analysis from the warping based algorithm, in Multi-scale Shift-map one can see that salient object should come from larger image sources, while non-salient region should come from smaller image sources in the stack. Shift-map has no control over which region in the output that contains the salient objects or non-salient objects, however it can prefer the appearance of certain 'good' labels. Since we want to preserve the image as much as possible, shifting to larger image sources should be preferred over smaller image sources. This can be achieved by assigning a larger data cost for shift to smaller image sources.

However, this direct approach will simply results in a normal Shift-map algorithm, since only the largest image source is shifted to. In case of extreme resizing (to very small size), some part of the image will be removed due to the size constraint, and hence smaller size image sources are preferred instead. A combination of image sources appearing in the output basically gives the warping effect, where some important regions are preserved while some are compressed to fit in the requirement size. To decide which regions should come from larger image sources and which regions should come from smaller image sources, we introduce a visual distortion measure. This mea-

sure determines whether a region in the image is distorted when the image is resized. Basically, a region with high distortion measure should be preserved as much as possible, while region with low distortion measure should come from smaller scale image sources, leaving more space for salient object in the output.

Each pixel p_i of the image i in the source stack is assigned an area cost $E_d(p_i) \propto 1/r_i$ where r_i is the resizing ratio of image i in the stack. Hence smaller image source pixels have smaller area cost. This means shifting to a smaller source does not cost much space in the output as in larger source. Besides, assuming a visual distortion measure is defined, each pixels in the source is also given a distortion cost

$$E_d(p_i) = D(p_i), \quad (3.8)$$

where $D(p_i)$ is the distortion cost of pixel p_i . The distortion measure is defined such that it gradually decreases from smaller scaled image to larger scaled image, depending on whether the pixel is in important region. Clearly, smooth regions will not have a large distortion cost, since scaling does not affect them a lot. A combination of distortion and area cost will then gives a good guide for the algorithm to decide which image sources to choose from. Among the same smooth regions in the image sources stack, smaller size sources are preferred since they have similar distortion cost and lower area cost. For important and salient regions, larger size image is better since the

distortion to shift to smaller size image is very high although they might have small area cost. The data term in the algorithm is then defined as:

$$E_d(p, l) = P(p, l) + \lambda * D(p, l), \quad (3.9)$$

where λ is a parameter to balance between the area cost and the distortion cost.

3.3.4 Smoothness constraints

In original Shift-map algorithm, the smoothness constraints is defined as the energy cost of mis-matching two neighboring pixels in the output. The basic understanding is that if two pixels that are not neighbors in the input are grouped together as neighbors in the output, artifacts may appear. The smoothness cost of any two neighboring pixels allocation is then defined as the difference between the current neighbor pixel and the original pixel in the input. In Multi-scale Shift-map, the original pixel here is detected not only from the original image in the stack, but from other scaled image sources depending on which label is used. It is noted that the same equation (3.5) is applicable here, since by applying the same label to the neighboring pixel will lead us to the correct image source. In the experiment, however we found that using this equation is too hard a constraint and often result in shifting to only one scaled image in the stack. Instead, we relax the constraint of

neighboring difference to:

$$R(S, T(p_1, l_1), T(p_2, l_2)) = \min([S(T(p_1, l_1)) - S(T(p_2, l_1))]^2, [S(T(p_1, l_2)) - S(T(p_2, l_2))]^2). \quad (3.10)$$

It is noted that in comparison with equation (3.5) we change the symbol I to S to represent the stack of images instead of one single image source. This equation allows a more relaxed constraint and allows easier transition between scaled image sources. We found that this relaxation does not result in any visible artifact and actually gives a very smooth transition between scaled image sources.

However, using only this smoothness cost is problematic in case of large object. The salient object may contain smooth region within itself, and result in the seam splitting two scaled sources cutting through the object and distorting the object shape. In order to prevent this, we propose using a smoothness cost that can preserve regions with high distortion value, i.e. require pixels in certain regions to stay together in the same source. The new smoothness cost E'_{sm} is then defined as follow:

$$\begin{aligned} E'_{sm}(T(p_1, l_1), T(p_2, l_2)) &= \text{Infinity} && \text{if } D(T(p_1, l_1), T(p_2, l_2)) > \theta \\ &= E_{sm} && \text{otherwise,} \end{aligned} \quad (3.11)$$

where E_{sm} is defined as in equation (3.4) and θ is the threshold parameter.

3.4 Experimental results and discussion

We conducted the experiment for Multi-scale Shift-map retargeting using the database provided by Rubinstein [31]. The proposed method is compared with Seam Carving (or SC) [3], Shift-map (or SM) [29], nonhomogeneous warping (or WARP) [38], Multi-operator (or MULTIOP) [33]. These are among the best algorithms based on the bench-mark provided by [31]. Our algorithm is fixed with 6 scaled images in the source stack including the original image, with scales 0.9, 0.8, 0.7, 0.6, 0.5 and 0.4 of the original size. For simplicity, the mentioned scales are only for horizontal resizing as we focused only on changing the width the input image. The smooth threshold in equation (3.11) is set to 10. Since our method suffers a similar problem with Shift-map in which the important region may not be recognized very well, a manual saliency map is used to retain important content in the image. To improve the speed of the algorithm, following Pritch et al [29] we allow only horizontal shift, and for each scale the number labels is only the difference between the size of the source and the output. With the help of the saliency map, the algorithm is able to retain the important region correctly in its original size. It is also interesting to note that Shift-map is able to combine different scales into 1 image with a clear border. This is different in comparison with warping-based methods where the transition from scale to scale must be continuous. In figure 3.6, the algorithm is able to compress the sky and cloud entirely to fit in the output image without changing the

Figure 3.6: Examples of retargeted battleship image from different methods. The input image is resized from 462×237 to 231×237 (50%). The last image illustrates which scaled source is used in which part of the output. The darker the color the larger the scale is, with black color denotes the original image.

size of the ship. This shows that the algorithm is able to compress and move regions in the input more freely to form the output. Another example can be seen as in figure 3.7, where by scaling different regions of the image differently, our algorithm can retain more content in the output in comparison with Shift-map algorithm.

In all examples mentioned previously, although the retargeted size is extreme i.e., 50% of the input image, the output size is still large enough to contain the salient object. Hence, we can see that the object is preserved and is shifted from the original source. In example shown in figure 3.8, we purposely resize the image to a smaller size in comparison to the important object size to force

Figure 3.7: Examples of retargeted pigeons image from different methods. The input image size is 320×240 , the output image size is 160×240 . Noted that building on the left is preserved in comparison to Shift-map

Figure 3.8: Retargeted output images when resized to different size. The original image size is 280×210 .

the algorithm to shift to a smaller source instead. It is interesting to see that when the output size is too small (from 210×210 to 190×210), the algorithm automatically switches to the next scaled source to ensure the main object is not cut off. It is also noted the building part (which is marked as salient) is also shifted from the next, smaller scaled source. It is also worth noticing that a part of the building is removed due to effect of Shift-map when the source itself is not small enough to fit in.

Although the experimental results look interesting, the algorithm still suffers from the same issue as the Shift-map where artifacts may arise when

important content is not preserved correctly. In fact, each retargeting needs a certain kind of guidance from saliency analysis to identify which region is important in order to achieve good result. If no guidance is given, our algorithm will simply pick a closest scaled source in the stack to shift to, the effect is close to linearly scale the image. In comparison to warping-based methods, our approach also has the limitation of the number of scaled sources in the stack. When a warping method can compress a region to any scale, the performance limits Multi-scale Shift-map to only a few scales. To increase the performance, we adopted a similar approach to Shift-map: the algorithm is performed in a pyramid manner, starting the retargeting process using a small scaled version of input image first and then using the result to infer the initial label mapping in the larger scale [29]. Also from our experiments, we observed that when the number of scaled sources is large enough, the algorithm can achieve the effect of warping close to warping-based methods. Furthermore, in case the scaled source does not fit exactly, the nature of Shift-map approach can remove some pixels in the non-important or smooth region to compensate.

3.5 Conclusion

In this chapter we have investigated a new framework which combines the power of the Shift-map approach and warping approach for image retargeting

application. Although Shift-map is a powerful algorithm with many potential usages, it lacks of the ability to incorporate scaling effect into the framework. This weakness of the frame-work often results in poor performance when the resizing ratio is small, causing important objects to disappear or be distorted. We attempted to tackle this problem by introducing a Multi-scale Shift-map, which makes use of multiple scales of the input. A new data term and smooth term are proposed in order to generalize the Shift-map framework to Multi-scale Shift-map correctly. The experimental results show that the new hybrid algorithm is able to combine the strength of both Shift-map and warping-based algorithms. In comparison with warping-based methods, the new algorithm can resize a region in the input more freely and remove unwanted objects if necessary. In the experiments, there are many examples show that our algorithm can achieve a better retargeting results given a good saliency map, especially in extreme resizing case. As the scope of the report for now is limited to the potential power of the approach, we conclude that the potential of the proposed framework is very promising. There are many important problems will need to be investigated in future in order to improve the algorithm. One of which is automatic important region analysis, which is also a crucial task for any image retargeting algorithms.

Chapter 4

Conclusions and future work

4.1 Conclusions

In this thesis we have investigated the use of sparse representation over redundant dictionary for saliency detection application. There is a vast and growing knowledge and research works regarding sparse representation in the literature, and we attempted to present a small fraction related to our topic. Some potential usage of sparse coding for saliency detection was discussed and experiments were conducted to evaluate the performance of the proposed approaches. The experiment demonstrated that using sparse coding methods such as K-SVD or efficient sparse coding provides sparser coefficients in comparison to standard methods like ICA, hence better resembling neurons

in the V1 visual cortex. Saliency map obtained by leveraging this advantage shows a superior performance in comparison with similar algorithms which use sparse representation provided by conventional methods.

We also proposed a new saliency algorithm which makes use of the statistical perspective of L1-minimization approach. Based on a dictionary assembled by concatenating input image patches directly, the algorithm measures saliency by the likelihood that a patch can be represented sparsely using other patches. Experimental results demonstrate that the proposed algorithm outperforms other state-of-the-art saliency algorithms. The framework of the algorithm is relatively simple yet flexible so that new helpful information can be integrated easily.

We also investigated a new framework which combines the power of the Shift-map approach and warping approach for image retargeting application. By stacking different scales of the input to form an input stack to the original Shift-map framework, the hybrid algorithm has the strength and the potential of both Shift-map and warping-based approaches. In order to generalize the framework to multi-scale inputs correctly, we introduced new data term and smooth term which can work directly across different layers of scaled inputs in the stack. The experiment has shown some very interesting results, in which the proposed algorithm is able to combine different regions in the input stacks to form the output. In some examples, this combination effect shows better performance than both Shift-map and other warping-based

methods. Unlike warping-based methods which use a continuous warping map, regions in the output can be compressed more freely. Furthermore, with the aid of a good saliency map, unimportant objects can be removed. This is an effect which is hard to achieve with the warping-based framework. Hence, by combining the strength of different approaches in retargeting, our algorithm has the potential of providing more flexible and better solutions.

4.2 Future work

The application of sparse representation over redundant dictionary for saliency detection is relatively new and hence there is a lot of rooms for future research in this direction. The proposed framework makes use of only the color and intensity information, while more meaningful information can be integrated to improve the result. For instance, spatial location of each image patch can be introduced to system to identify compact salient object correctly. Segmentation information can help the algorithm to tackle difficult regions such as object boundaries. In a broader scope, how to apply sparse coding algorithm for natural image data is a new and fast moving research area in computer vision.

The proposed Multi-scale Shift-map framework is able to introduce warping effect to the original Shift-map framework. However, introducing multi-scale inputs means the graph cut algorithm has to deal with increasing number of

possible labels. How to choose the scaled sources wisely in order to improve the speed of the algorithm as well as the retargeted result. Importance content is also a crucial aspect for any retargeting algorithm, how to choose a proper saliency map to guide the algorithm will be an open problem that needs to be solved.

Bibliography

- [1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, June 2009.
- [2] 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, 2006.
- [3] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics*, 26(3):10, July 2007.
- [4] E. Bingham and A. Hyvärinen. A fast fixed-point algorithm for independent component analysis of complex valued signals. *International journal of neural systems*, 10(1):1–8, February 2000.
- [5] N. Bruce and J. Tsotsos. Saliency based on information maximization. *Advances in neural information processing systems*, 18:155–162, 2006.

- [6] A.M. Bruckstein, D.L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.
- [7] Emmanuel Candes and Terence Tao. Error Correction via Linear Programming. *Annual IEEE Symposium on Foundations of Computer Science*, pages 668 – 681, 2005.
- [8] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1999.
- [9] David L. Donoho and Yaakov Tsaig. Fast Solution of L1-norm Minimization Problems When the Solution May be Sparse. *IEEE Transactions on Information Theory*, 54(11):1–45, 2006.
- [10] B. Efron, T. Hastie, and I. Johnstone. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.
- [11] M. Elad, M.A.T. Figueiredo, and Y. Ma. On the Role of Sparse and Redundant Representations in Image Processing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2010.
- [12] Y. C. Eldar, P. Kuppinger, and B. Helmut. Compressed Sensing of Block-Sparse Signals : Uncertainty Relations and Efficient Recovery. *IEEE Transactions on Signal Processing*, 2009.

- [13] E. Elhamifar. Clustering disjoint subspaces via sparse representation. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1926 – 1929, 2010.
- [14] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797. Ieee, June 2009.
- [15] P.J. Garrigues and B.A. Olshausen. Group sparse coding with a laplacian scale mixture prior. *Advances in Neural Information Processing Systems*, 23:1–9, 2010.
- [16] Y. Guo, F. Liu, and J. Shi. Image Retargeting Using Mesh Parametrization. *IEEE Transactions on Multimedia*, 11(5):1–14, 2009.
- [17] X. Hou and L. Zhang. Saliency Detection: A Spectral Residual Approach. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.
- [18] X. Hou and L. Zhang. Dynamic visual attention: Searching for coding length increments. *Advances in neural information processing systems*, 21(800):681–688, 2008.
- [19] Y. Hu and D. Rajan. Hybrid shift map for video retargeting. *Computer Vision and Pattern Recognition*, pages 577–584, 2010.

- [20] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
- [21] E. Kav-Venaki and S. Peleg. Feedback Retargeting. *Media Retargeting Workshop at ECCV 2010*, 2010.
- [22] S. Kim. An Interior-Point Method for Large-Scale Logistic Regression. *Journal of Machine Learning Research*, 8:1519–1555, 2007.
- [23] S. Kim, K. Koh, M. Lustig, and S. Boyd. An interior-point method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–617, 2007.
- [24] H. Lee, A. Battle, R. Raina, and A.Y. Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007.
- [25] X. Li. Data-Driven Approach for Bridging the Cognitive Gap in Image Retrieval. *IEEE International Conference on Multimedia and Expo*, pages 2231–2234, 2004.
- [26] Y. Li, Y. Zhou, L. Xu, X. Yang, and J. Yang. Incremental Sparse Saliency Detection. *IEEE International Conference on Image Processing*, 2009.

- [27] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE transactions on image processing*, 17(1):53–69, January 2008.
- [28] Y. C. Pati and R. Rezaiifar. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, page 1, 1993.
- [29] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. *Proceedings of the Twelfth IEEE International Conference on Computer Vision*, 721, 2009.
- [30] Z. Ren, Y. Hu, L.T. Chia, and D. Rajan. Improved saliency detection based on superpixel clustering and saliency propagation. In *Proceedings of the international conference on Multimedia*, number 2, pages 1099–1102. ACM, 2010.
- [31] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. A comparative study of image retargeting. *ACM Transactions on Graphics*, 29(6):160, 2010.
- [32] M. Rubinstein, A. Shamir, and S. Avidan. Improved seam carving for video retargeting. *ACM Transactions on Graphics*, 27(3):1, August 2008.

- [33] M. Rubinstein, A. Shamir, and S. Avidan. Multi-operator media retargeting. *ACM Transactions on Graphics*, 28(3):1, July 2009.
- [34] X. Sun, H. Yao, R. Ji, P. Xu, X. Liu, and S. Liu. Saliency detection based on short-term sparse representation. In *IEEE International Conference on Image Processing*, pages 1101–1104. IEEE, 2010.
- [35] R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 1996.
- [36] Y. Tsaig and D. Donoho. Breakdown of equivalence between the minimal 11-norm solution and the sparsest solution. *Signal Processing*, 86(3):533–548, March 2006.
- [37] W. E. Vinje and J. L. Gallant. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–6, February 2000.
- [38] Y. Wang. Optimized scale-and-stretch for image resizing. *ACM Transactions on Graphics*, 27(5):1, December 2008.
- [39] J. Wright, J. Mairal, G. Sapiro, and T. Huang. Sparse Representation for Computer Vision and Pattern Recognition. *Proceedings of the IEEE*, 98(6):1031–1044, June 2010.
- [40] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–27, February 2009.

- [41] Y. Tsaig. *Sparse solution of underdetermined linear systems: algorithms and applications*. PhD thesis, Stanford, 2007.
- [42] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [43] S. Zhang and S. Mallat. Matching pursuit with time-frequency dictionaries. *IEEE Trans. Signal Processing*, 41:3397–3415, 1993.