ABSTRACT

Title of dissertation: VISUAL TRACKING AND ILLUMINATION RECOVERY VIA SPARSE REPRESENTATION

Xue Mei, Doctor of Philosophy, 2009

Dissertation directed by: Professor David W. Jacobs
Department of Computer Science
And
Professor Gang Qu
Department of Electrical and Computer Engineering

Compressive sensing, or sparse representation, has played a fundamental role in many fields of science. It shows that the signals and images can be reconstructed from far fewer measurements than what is usually considered to be necessary. Sparsity leads to efficient estimation, efficient compression, dimensionality reduction, and efficient modeling. Recently, there has been a growing interest in compressive sensing in computer vision and it has been successfully applied to face recognition, background subtraction, object tracking and other problems. Sparsity can be achieved by solving the compressive sensing problem using ℓ_1 minimization. In this dissertation, we present the results of a study of applying sparse representation to illumination recovery, object tracking, and simultaneous tracking and recognition.

Illumination recovery, also known as inverse lighting, is the problem of recovering an illumination distribution in a scene from the appearance of objects located in the scene. It is used for Augmented Reality, where the virtual objects match the existing

image and cast convincing shadows on the real scene rendered with the recovered illumination. Shadows in a scene are caused by the occlusion of incoming light, and thus contain information about the lighting of the scene. Although shadows have been used in determining the 3D shape of the object that casts shadows onto the scene, few studies have focused on the illumination information provided by the shadows. In this dissertation, we recover the illumination of a scene from a single image with cast shadows given the geometry of the scene. The images with cast shadows can be quite complex and therefore cannot be well approximated by low-dimensional linear subspaces. However, in this study we show that the set of images produced by a Lambertian scene with cast shadows can be efficiently represented by a sparse set of images generated by directional light sources. We first model an image with cast shadows as composed of a diffusive part (without cast shadows) and a residual part that captures cast shadows. Then, we express the problem in an ℓ_1-regularized least squares formulation, with nonnegativity constraints (as light has to be nonnegative at any point in space). This sparse representation enjoys an effective and fast solution, thanks to recent advances in compressive sensing. In experiments on both synthetic and real data, our approach performs favorably in comparison to several previously proposed methods.

Visual tracking, which consistently infers the motion of a desired target in a video sequence, has been an active and fruitful research topic in computer vision for decades. It has many practical applications such as surveillance, human computer interaction, medical imaging and so on. Many challenges to design a robust tracking algorithm come from the enormous unpredictable variations in the target, such as deformations, fast motion, occlusions, background clutter, and lighting changes. To tackle the challenges posed by

tracking, we propose a robust visual tracking method by casting tracking as a sparse approximation problem in a particle filter framework. In this framework, occlusion, noise and other challenging issues are addressed seamlessly through a set of trivial templates. Specifically, to find the tracking target at a new frame, each target candidate is sparsely represented in the space spanned by target templates and trivial templates. The sparsity is achieved by solving an ℓ_1-regularized least squares problem. Then the candidate with the smallest projection error is taken as the tracking target. After that, tracking is continued using a Bayesian state inference framework in which a particle filter is used for propagating sample distributions over time. Three additional components further improve the robustness of our approach: 1) a velocity incorporated motion model that helps concentrate the samples on the true target location in the next frame, 2) the nonnegativity constraints that help filter out clutter that is similar to tracked targets in reversed intensity patterns, and 3) a dynamic template update scheme that keeps track of the most representative templates throughout the tracking procedure. We test the proposed approach on many challenging sequences involving heavy occlusions, drastic illumination changes, large scale changes, non-rigid object movement, out-of-plane rotation, and large pose variations. The proposed approach shows excellent performance in comparison with four previously proposed trackers. We also extend the work to simultaneous tracking and recognition in vehicle classification in IR video sequences. We attempt to resolve the uncertainties in tracking and recognition at the same time by introducing a static template set that stores target images in various conditions such as different poses, lighting, and so on. The recognition results at each frame are propagated to produce the final result for the whole video. The tracking result is evaluated at each frame and low confidence in track-

ing performance initiates a new cycle of tracking and classification. We demonstrate the robustness of the proposed method on vehicle tracking and classification using outdoor IR video sequences.

VISUAL TRACKING AND ILLUMINATION RECOVERY VIA SPARSE REPRESENTATION

by

Xue Mei

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:
Professor Gang Qu, Chair/Advisor
Professor David W. Jacobs, Co-Advisor
Professor Nuno Martins
Professor Larry Davis
Professor Haibin Ling

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this work. Without their valuable support, I would not go this far and have a chance to finish the work I love to be working on. First and foremost, I would like to thank my advisors, Prof. Gang Qu and Prof. David W. Jacobs, whose guidance and support over the last few years have been of great help. I am grateful of being given the opportunity to work on these challenging and interesting problems. Without their valuable comments and research expertise, this dissertation would have been a distant dream.

I am deeply indebted to my colleague, friend, office mate, and most important, mentor, Dr. Haibin Ling, whose stimulating suggestions and encouragement helped me survive during the hardest time in my life. It is him who kept telling me that there is light at the end of the tunnel and made me understand that I can make a difference if I have the dream and keep working hard.

I am grateful to the committee members, Prof. Nuno Martins and Prof. Larry Davis, for sparing their valuable time to be in my committee. Their insightful comments on my research have greatly improved the quality of this dissertation. I would like to thank Prof. Francois Guimbretiere, for his permission to access the 3D printer for my illumination recovery work.

I would like to thank my former advisor, Prof. Rama Chellappa, who brought me to the U.S. and gave me the opportunity to be studying in this great university. I am deeply indebted to his support and help for the first three and half years during my graduate study.

I would like to thank Sameer Shirdhonkar for providing me with the SDP code and

Chunyuan Liao for helping me with the 3D printing.

My colleagues at the Center for Automation Research have enriched my graduate life in many ways and deserve a special mention. Dr. Kevin S. Zhou helped me start-off by giving me the first tracking code to work on. My interaction with Yang Ran, Feng Guo, Arunkumar Mohananchettiar, Yang Yu, Gang Qian, Ashwin Sankaranarayanan, Mahesh Ramachandran, Hao Wu, Narayanan Ramanathan, Ashok N. Veeraraghavan, Jie Shao, Zhanfeng Yue, and Qinfen Zheng has been very fruitful.

I would like to thank many friends like Ming Luo, Cheng Shao, Lei Zhang, Chao Wu, Jiao Yu, Jing Yang, Juanjuan Xiang and many more in the University of Maryland, College Park.

I also want to thank my mentor Dr. Fatih Porikli from Mitsubishi Electric Research Laboratories, and Dr. Peiya Liu from Siemens Corporate Research for their support and help while I was doing an internship there.

Especially, I would like to give my special thanks to my daughter, Melinda, who brings endless happiness to me and encourages me to work hard to give her a better life.

It is impossible to remember all, and I apologize to those I have inadvertently left out.

Table of Contents

List of Figures

Chapter 1

Introduction

Compressive sensing has played a fundamental role in many fields of science such as signal processing, image processing, and so on. Recently, many applications using compressive sensing have been successfully applied in computer vision such as face recognition, background subtraction, and so on. This dissertation is the first to apply compressive sensing to the study of illumination recovery and object tracking.

The whole dissertation is organized as follows. The first chapter gives an introduction to compressive sensing and its solution through ℓ_1 minimization, the development in illumination recovery and visual tracking, and particle filters in the context of visual tracking. The second chapter presents the study on illumination recovery using compressive sensing. The lighting is approximated using spherical harmonics and a small number of directional sources selected using ℓ_1 minimization. It achieves favorable experimental results on both synthetic and real data in terms of speed and accuracy compared with other illumination recovery methods. The third chapter describes the visual tracking work using compressive sensing. Tracking is cast as a sparse representation problem in a particle filter framework. Occlusion, noise and other challenging issues are addressed seamlessly through a set of trivial templates. It shows promising experimental results on numerous challenging video sequences compared with four other state-of-the-art trackers. The last chapter concludes the dissertation and points to future directions.

1.1 Compressive Sensing

Compressive sensing has received a great deal of attentions in recent years [16, 18, 34, 147]. The problem is to exploit the compressibility and sparsity of the true signal and use a lower sampling frequency than the Shannon-Nyquist rate. By transforming a large image into a small number of appropriate basis elements and then coding only the important expansion coefficients, we convert a high-resolution image into a relatively small bit stream. The image can then be saved or transferred efficiently. This is especially useful in today's life where a good compression algorithm is necessary to meet the high demands of bandwidth. Compressive sensing is an ℓ_0 minimization problem that is usually hard to solve. Recent studies [16, 33] show that, under very flexible conditions, the ℓ_0 minimization can be reduced to ℓ_1 minimization that further results in a convex optimization [13], which can be solved efficiently. In this section, we give a brief introduction to compressive sensing. An extensive list of compressive sensing resources can be found at the compressive sensing resources web site [25].

1.1.1 Theoretical Background

Many natural signals have concise representations when expressed in a convenient basis. Mathematically speaking, we have a vector y represented as a linear combination of a basis and written in a linear system as follows:

$$y = Ax + \epsilon \tag{1.1}$$

where $x \in \mathbb{R}^n$ is a k-sparse unknown discrete vector that contains at most $k \leq n$ nonzero elements, $A \in \mathbb{R}^{m \times n}$ is the measurement matrix, $y \in \mathbb{R}^m$ is the vector of measurements,

and $\epsilon \in \mathbb{R}^m$ is the noise.

Obviously, if $m \geq n$ and the columns of A are linearly independent, the system is overdetermined and x can be faithfully reconstructed from the measurements, without any prior knowledge of the system. The unique solution of x is obtained by solving the least squares problem of minimizing the quadratic error $||Ax - y||_2^2$, where $||u||_2$ denotes the ℓ_2 norm of u. However, if the number of measurements, m, is not large enough compared to n, the system is underdetermined and simple least-squares regression leads to over-fit.

1.1.2 Reconstruction using ℓ_2 Minimization

To reconstruct the unknown signal x from the measurements y, a standard technique to prevent over-fitting is ℓ_2 or Tikhonov regularization [96], which can be written as

$$\arg\min_x ||Ax - y||_2^2 + \lambda ||x||_2^2 \tag{1.2}$$

where $||x||_2 = (\sum_{k=1}^{N} x_k^2)^{1/2}$ denotes the ℓ_2 norm of x and $\lambda > 0$ is the regularization parameter. The Tikhonov regularization problem or ℓ_2-regularized least-squares problem (LSP) has the analytic solution.

$$x^{\ell_2} = (A^T A + \lambda I)^{-1} A^T y \tag{1.3}$$

The solutions to the Tikhonov regularization problem can be computed by direct methods, which require $O(n^3)$ flops, when no structure is exploited. The solution can also be computed by applying iterative methods to the linear system of equations $(A^T A + \lambda I)x = A^T y$. Iterative methods are efficient especially when there are fast algorithms the matrix-vector multiplications with the data matrix A and its transpose A^T.

1.1.3 Reconstruction using ℓ_1 Minimization

In ℓ_1 regularized least squares, we substitute a sum of absolute values for the sum of squares used in Tikhonov regularization, to obtain

$$\arg\min_x ||Ax - y||_2^2 + \lambda||x||_1 \tag{1.4}$$

where $||x||_1 = \sum_{k=1}^{N} |x_k|$ denotes the ℓ_1 norm of x and $\lambda > 0$ is the regularization parameter. This problem always has a solution, though not necessarily unique. ℓ_1-regularized least squares (LS) typically yields a sparse vector x, which has relatively few nonzero coefficients. In contrast, the solution to the Tikhonov regularization problem generally has all coefficients nonzero. There is no analytic formula or expression for the optimal solution to the ℓ_1-regularized LSP. Its solution must be computed numerically. The objective function in the ℓ_1 regularized LSP is convex but not differentiable, so solving it is more of a computational challenge than solving the ℓ_2-regularized LSP.

The sparse solution enforced by ℓ_1 minimization can be solved fairly efficiently using either the simplex algorithm [29] or standard convex optimization [13] methods such as interior-point methods [70, 97, 136, 141]. Standard interior-point methods are implemented in general purpose solvers which can readily handle small and medium size problems. Specialized interior-point methods can scale to large problems, as demonstrated in [65, 70]. The optimization strategy that solves (1.4) is also called Basis Pursuit (BP) [21, 33]. An alternative algorithm called Orthogonal Matching Pursuit (OMP) [130] to BP offers faster and easier implementation while achieving comparable results for signal recovery problems.

Recently, several researchers have proposed homotopy methods and variants for

solving ℓ_1-regularized LSPs [35, 40, 55, 104, 115]. Using the piecewise linear property of the regularization path, path-following methods can compute efficiently the entire solution path in an ℓ_1 regularized LSP. Other recently developed computational methods include coordinate-wise descent methods [42], Bregman iterative regularization based methods [105, 143], sequential subspace optimization methods [95]. Bayesian compressive sensing [63] is extended to decode measurements of multiple signals simultaneously by using hierarchical Bayesian models [64].

Our implementation solves the ℓ_1-regularized least squares problem via an interior-point method based on [70]. The method uses the preconditioned conjugate gradients (PCG) algorithm to compute the search direction and the run time is determined by the product of the total number of PCG steps required over all iterations and the cost of a PCG step. We use the code from [27] for the minimization task in the following chapters.

1.1.4 Error Bounding

The accuracy of the reconstruction depends on two factors and the error is bounded as follows [19]:

$$||\tilde{x} - x||_2 \leq C_0 * ||x - x_k||_1/\sqrt{k} + C_1 * \epsilon \tag{1.5}$$

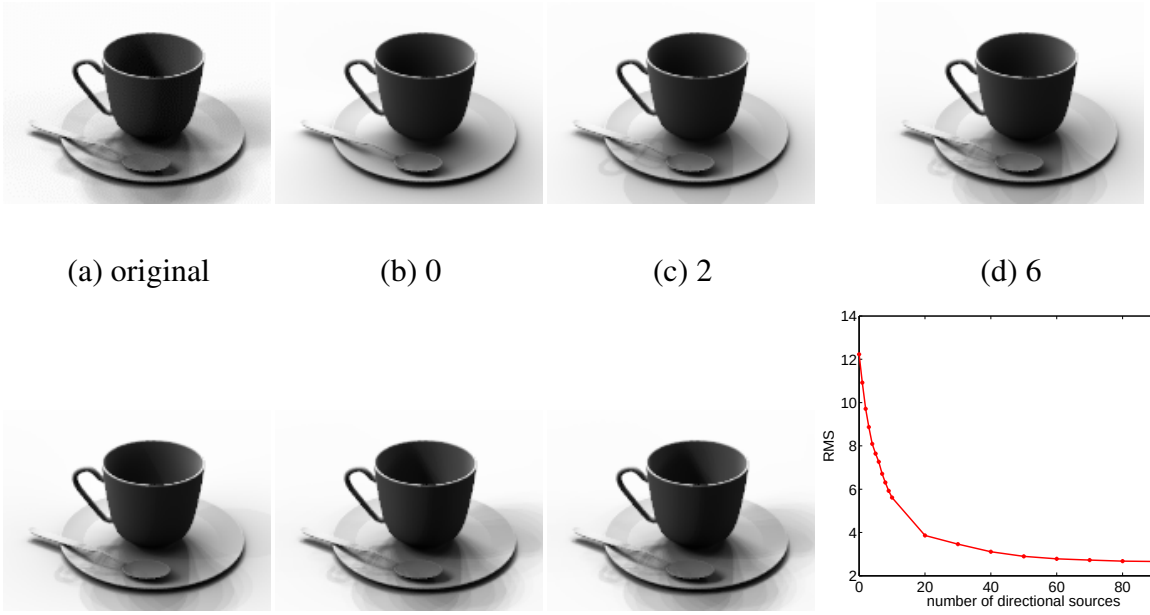
where $\tilde{x}$ is the solution, x_k is the vector x with all but the largest k elements set to 0. C_0 and C_1 are two constants. ϵ bounds the noise of the data. The first term measures the sparsity of the signal. If x is k sparse, then $x = x_k$ and thus the first term is 0. If x is more than k sparse, we can decrease the first term error by increasing the number of nonzero elements in x. The second term measures the noise of the data, where $||A\tilde{x} - y||_2 \leq \epsilon$.

This measurement error is proportional to the noise level. Since compressive measurements y does not appear in both terms, both errors are not amplified by the compressive measurements. The number of measurements required to successfully reconstruct a k-term approximation of a signal of length n has to be $C * k\log(n/k)$, where C is some constant depending on the measurement matrix A.

1.1.5 Compressibility vs. Sparsity

While the previous discussion assumed the signal is exactly k-sparse, most real world signals are not necessarily sparse, but rather compressible in that they have a lot of small coefficients. If the sorted magnitude of the x_i decay quickly, then x is well approximated by the largest k coefficients in x by throwing away a large fraction of the coefficients without much loss. A good example is the JPEG lossy coder, where coefficients are quantized and small ones become zeros and are thrown away. Compressive sensing also works well in recovering the best k-term approximation to the original signal [18].

Figure 1.1 shows the original image (a), images rendered with recovered lighting using no directional lights (DS) (b), 2 DS (c), 6 DS (d), 10 DS (e), 60 DS (f), 100 DS (g), and error plot (h), respectively. The error plot shows the root-mean-square (RMS) error vs. the number of directional lights used in approximation. The more directional lights we use, the better approximation results we achieve. However, after a certain number of directional lights, the error remains almost constant, and the perceptual loss is hardly noticeable from the rendered images with the recovered lighting (Figure 1.1 (f) and (g)).



(a) original (b) 0 (c) 2 (d) 6

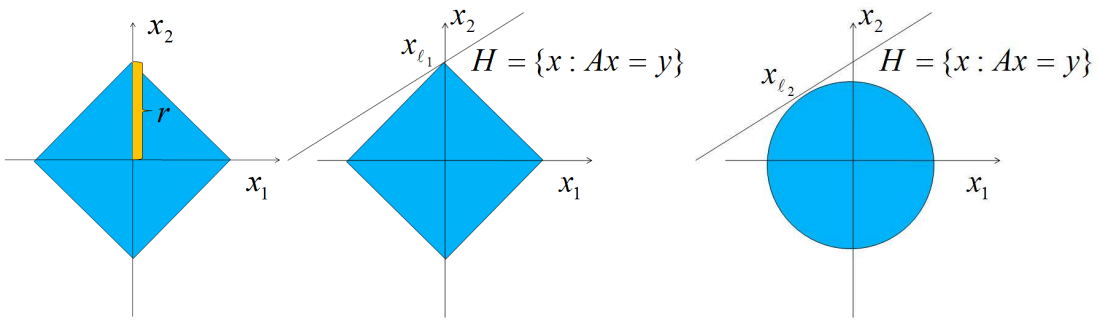
(e) 10 (f) 60 (g) 100 (h) error plot

Figure 1.1: After a certain number of directional lights, the error remains almost constant, and the perceptual loss is hardly noticeable from the rendered images with the recovered lighting.

1.1.6 The Geometry of ℓ_1 minimization

We can get some geometric intuition for why ℓ_1 is an effective substitute for sparsity by giving a 2D illustration in Figure 1.2 (essentially due to [113, 32]). A 3D illustration can be found in [137]. Figure 1.2 (a) illustrates the ℓ_1 ball in $\mathbb{R}^2$ of radius r. Note that it is anisotropic. While the standard Euclidean ℓ_2 ball is spherical and thus completely isotropic. (b) diagrams the ℓ_1 recovery program: the point labeled x_{ℓ_1} is a sparse vector which only has one nonzero element. The line labeled H is the set of all x that share the same measurement value y.

The task for the equation is to pick out the point on the line with minimum ℓ_1 norm. To visualize how the equation accomplishes this, imagine taking an ℓ_1 ball of tiny radius



(a) (b) (c)

Figure 1.2: Modeled on [113]. Geometry of ℓ_1 recovery. (a) ℓ_1 ball with radius r. The gray region contains all $x \in \mathbb{R}^2$ such that $|x_1| + |x_2| \geq r$. (b) Solving the ℓ_1 minimization problem allows us to recover a sparse x_{ℓ_1} from $\Phi x = y$, as the anisotropy of the ℓ_1 ball favors sparse vectors. (c) Minimizing the ℓ_2 norm does not recover x_{ℓ_1}, since the ℓ_2 ball is isotropic, the solution x_{ℓ_2} will in general not be sparse at all.

and gradually expanding it until it bumps into H. This first point of intersection is by definition the vector that solves the equation. The combination of the anisotropy of the ℓ_1 ball and the flatness of the space H results in this intersection occurring at one of the points, precisely where sparse signals are located.

Compare this to what would happen if we replaced the ℓ_1 norm with the ℓ_2 norm which would make the recovery a least-squares problem. Figure 1.2 (c) replaces the diamond shaped ℓ_1 ball with the spherical and perfectly isotropic ℓ_2 ball. We can see that the point of first intersection of H and the expanding ℓ_2 ball does not have to be sparse at all. In high dimensions this difference becomes very dramatic. Despite the seemingly innocuous difference in the definitions of the ℓ_1 and ℓ_2 norms, they are totally different creatures.

1.1.7 Applications of Compressive Sensing

Since its inception just a few years ago, compressive sensing has been popular in signal processing due to its applicability to a wide range of diverse signal processing problems, from source separation [30], de-noising [41], coding [46], to sampling [17]. Other than signal processing, it is gaining popularity in a wide range of applications in video processing [82], medical imaging [80, 131, 109], bio-sensing [28, 123], and compressive imaging [39, 44, 134].

Within the computer vision community, compressive sensing has been applied to a variety of problems. In [137], face recognition is cast as the problem of classifying among multiple linear regression models and the coefficients from training samples encode the identity of the face. A new dynamic group sparsity learning algorithm is proposed in [58] and successfully applied to background subtraction in videos. Background subtraction using compressive sensing is also studied in [20]. In [89], illumination distribution is approximated from an image with cast shadows using directional light sources and finds the best directional sources through ℓ_1 minimization. In [90], tracking is cast as a sparse representation on the subspace spanned by the target template set and trivial template set. In [81], a novel framework is introduced for using learned sparse image representation in local classification tasks and leads to state-of-the-art segmentation result. In [47], a new method named compressive structured light is proposed for recovering inhomogeneous participating media. In [57], a technique for obtaining transformation-invariant image sparse representation is introduced and simultaneously recovering the sparse representation of a target image and the image plane transformation between the target and the

model image. There is also work related to lighting in computer graphics using compressive sensing [107, 121].

1.1.8 Solution Through Truncated Newton Interior-Point Method

In this subsection, we briefly review a truncated Newton interior-point method in [70]. In the following illumination recovery and visual tracking work, we use this method to solve the ℓ_1 minimization.

The ℓ_1-regularized LSP in (1.4) can be transformed to an equivalent convex Quadratic Problem (QP), with linear inequality constraints.

$$\begin{aligned} &\arg\min_x ||Ax - y||_2^2 + \lambda \sum_{i=1}^{n} u_i \\ \text{subject to } &-u_i \leq x_i \leq u_i,\ i = 1, 2, \cdots, n \end{aligned} \tag{1.6}$$

where the variables are $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$.

1.1.8.1 Dual Problem and Suboptimality Bound

We derive a Lagrange dual of the ℓ_1-regularized LSP (1.4). We start by introducing a new variable $z \in \mathbb{R}^m$, as well as new equality constraints $z = Ax - y$, to obtain the equivalent problem

$$\begin{aligned} &\text{minimize } z^T z + \lambda ||x||_1 \\ &\text{subject to } z = Ax - y \end{aligned} \tag{1.7}$$

Associating dual variables $\nu_i \in \mathbb{R}$, $i = 1, 2, \cdots, m$ with the equality constraints,

the Lagrangian is

$$L(x, z, \nu) = z^T z + \lambda ||x||_1 + \nu^T (Ax - y - z) \tag{1.8}$$

The Lagrange dual of (1.7) is therefore

$$\begin{aligned} &\text{maximize } G(\nu) \\ &\text{subject to } |(A^T \nu)_i| \leq \lambda \end{aligned} \tag{1.9}$$

where the dual objective $G(\nu)$ is

$$G(\nu) = -\frac{1}{4}\nu^T \nu - \nu^T y \tag{1.10}$$

The optimal values of the primal (1.4) and dual (1.9) are equal since the primal problem (1.4) satisfies Slater's condition.

An important property of the ℓ_1-regularized LSP (1.4) is that from an arbitrary x, we can derive an easily computed bound on the suboptimality of x, by constructing a dual feasible point

$$\begin{aligned} \nu &= 2s(Ax - y), \\ s &= \min \frac{\lambda}{|2((A^T Ax)_i - 2y_i)|} \; i = 1, 2, \cdots, m \end{aligned} \tag{1.11}$$

The point ν is dual feasible, so $G(\nu)$ is a lower bound on the optimal value of the ℓ_1-regularized LSP (1.4).

The difference between the primal objective value of x and the associated lower bound $G(\nu)$ is called the *duality gap*. We use η to denote the gap

$$\eta = ||Ax - y||_2^2 + \lambda ||x||_1 - G(\nu) \tag{1.12}$$

The duality gap is always nonnegative by weak duality, and x is not more than η-suboptimal. At an optimal point, the duality gap is zero, i.e., strong duality holds.

1.1.8.2 A Custom Interior-Point Method

We start by defining the logarithmic barrier for the bound constraints $-u_i \leq x_i \leq u_i$ in (1.6)

$$\Phi(x,u) = -\sum_{i=1} \log(u_i + x_i) - \sum_{i=1} \log(u_i - x_i) \tag{1.13}$$

defined over domain$\Phi = \{(x,u) \in \mathbb{R}^n \times \mathbb{R}^n | |x_i| < u_i, i = 1, 2, \cdots, n\}$. The central path consists of the unique minimizer $(x^*(t), u^*(t))$ of the convex function

$$\phi_t(x,u) = t||Ax - y||_2^2 + t\sum_{i=1}^{n} \lambda u_i + \Phi(x,u) \tag{1.14}$$

as the parameter t varies from 0 to ∞. With each point $(x^*(t), (u^*(t)$ on the central path we associate $\nu^*(t) = 2(Ax^*(t) - y)$, which can be shown to be dual feasible.

In the primal interior-point method, we compute a sequence of points on the central path, for an increasing sequence of values of t, starting from the previously computed central point. In the primal barrier method, Newton's method is used to minimize ϕ_t, i.e., the search direction is computed as the exact solution to the Newton system.

$$H \begin{bmatrix} \Delta x \\ \\ \Delta u \end{bmatrix} = -g \tag{1.15}$$

where $H = \Delta^2\phi_t(x,u) \in \mathbb{R}^{2n \times 2n}$ is the Hessian and $g = \Delta\phi_t(x,u) \in \mathbb{R}^{2n}$ is the gradient at the current iteration (x,u).

For a large ℓ_1-regularized LSP, solving the Newton system (1.15) exactly is not computationally practical. We need to find a search direction which gives a good trade-off of computational effort versus the convergence rate it provides. The algorithm for the ℓ_1-regularized LSP is listed in Algorithm 1.

Algorithm 1 Truncated Newton method for ℓ_1-regularized LSP

Given relative tolerance $\epsilon > 0$.

Initialize. $t = 1/\lambda, x = 0, u = 1$.

repeat

  Compute the search direction $(\Delta x, \Delta u)$ as an approximate solution to the Newton system (1.15).

  Compute the step size s by backtracking line search.

  Update the iterate by $(x, u) = (x, u) + s(\Delta x, \Delta u)$.

  Normalize $\tilde{I}_k$ such that $||\tilde{I}_k||_2 = 1$.

  Construct a dual feasible point ν from (1.11).

  Evaluate the duality gap η from (1.12).

  Update t.

until $\eta/G(\nu) \leq \epsilon$

As a stopping criterion, the method uses the duality gap divided by the dual objective value. By weak duality, the ratio is an upper bound on the relative suboptimality

$$\frac{f(x) - p^*}{p^*} \leq \frac{\eta}{G(\nu)} \tag{1.16}$$

where p^* is the optimal value of the ℓ_1-regularized LSP (1.4) and $f(x)$ is the primal objective computed with the point x. Therefore, the method solves the problem to guaranteed relative accuracy ϵ.

Given the search direction $(\Delta x, \Delta u)$, the new point is $(x, u) + s(\Delta x, \Delta u)$. s is the step size. In the backtracking line search, the step size is taken as $s = \beta^k$, where $k \geq 0$ is

the smallest integer that satisfies

$$\phi_t(x + \beta^k \Delta x, u + \beta^k \Delta u) \leq \phi_t(x, u) + \alpha \beta^k \Delta \phi_t(x, u)^T [\Delta x, \Delta u] \tag{1.17}$$

where $\alpha \in (0, 1/2)$ and $\beta \in (0, 1)$ are algorithm parameters.

In the primal barrier method, the parameter t is held constant until ϕ_t is minimized. For faster convergence, we can update the parameter t at each iteration based on the current duality gap. We use the update rule

$$t = \begin{cases} \max\{\mu \min\{2n/\eta, t\}, t\}, & s \geq s_{min} \\ t, & s < s_{min} \end{cases} \tag{1.18}$$

where $\mu > 1$ and $s_{min} \in (0, 1)$ are parameters to be chosen.

1.1.8.3 Search Direction via PCGs

We can find compact representations of the Hessian and gradient. The Hessian can be written as

$$H = t\Delta^2 ||Ax - y||_2^2 + \Delta^2 \Phi(x, u) = \begin{bmatrix} 2tA^T A + D_1 & D_2 \\ D_2 & D_1 \end{bmatrix} \tag{1.19}$$

where

$$\begin{aligned} D_1 &= diag(\frac{2(u_1^2 + x_1^2)}{(u_1^2 - x_1^2)^2}, \cdots, \frac{2(u_n^2 + x_n^2)}{(u_n^2 - x_n^2)^2}) \\ D_2 &= diag(\frac{-4u_1 x_1}{(u_1^2 - x_1^2)^2}, \cdots, \frac{-4u_n x_n}{(u_n^2 - x_n^2)^2}) \end{aligned} \tag{1.20}$$

Here, we use $diag(A_1, \cdots, A_p)$ to denote the diagonal matrix with diagonal blocks $A_1, \cdots, A_p$. The Hessian H is symmetric and positive definite. The gradient can be written as

$$g = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \in \mathbb{R}^{2n} \tag{1.21}$$

where

$$g_1 = \Delta_x \phi_t(x, u) \tag{1.22}$$

$$= 2tA^T(Ax - y) + \begin{bmatrix} 2x_1/(u_1^2 - x_1^2) \\ 2x_n/(u_n^2 - x_n^2) \end{bmatrix} \in \mathbb{R}^n \tag{1.23}$$

$$g_2 = \Delta_u \phi_t(x, u) \tag{1.24}$$

$$= t\lambda - \begin{bmatrix} 2u_1/(u_1^2 - x_1^2) \\ 2u_n/(u_n^2 - x_n^2) \end{bmatrix} \in \mathbb{R}^n \tag{1.25}$$

We compute the search direction approximately, applying the PCG algorithm to the Newton system (1.15). The PCG algorithm uses a preconditioner $P \in \mathbb{R}^{2n \times 2n}$, which is symmetric and positive definite.

1.1.8.4 ℓ_1-Regularized LSPs with Nonnegativity Constraints

The ℓ_1-Regularized LSPs with Nonnegativity Constraints is formulated as follows

$$\arg\min_x ||Ax - y||_2^2 + \lambda \sum_{i=1}^{n} x_i$$
$$\text{subject to } x_i \geq 0, i = 1, 2, \cdots, n \tag{1.26}$$

The associated centering problem is to minimize the weighted objective function augmented by the logarithmic barrier for the constraints $x_i \geq 0$

$$\arg\min_x t||Ax - y||_2^2 + t\lambda \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \log x_i \tag{1.27}$$

The Newton system for the centering problem is

$$(2tA^TA + D)\Delta x = -g \tag{1.28}$$

where

$$D = \begin{bmatrix} \frac{1}{x_1^2} \\ \vdots \\ \frac{1}{x_n^2} \end{bmatrix} \tag{1.29}$$

$$g = 2tA^T(Ax - y) + \begin{bmatrix} t\lambda_1 - \frac{1}{x_1} \\ \vdots \\ t\lambda_n - \frac{1}{x_n} \end{bmatrix} \tag{1.30}$$

The diagonal preconditioner of the form

$$P = diag(2tA^TA) + D \tag{1.31}$$

works well for this problem, especially when the optimal solution is sparse.

1.2 Illumination Recovery

Illumination recovery, also known as inverse lighting, is to recover an illumination distribution in a scene from the appearance of objects located in the scene. There has been a series of work aimed at understanding the complexity of the set of images produced by Lambertian objects lit by environment maps. It is shown in [122, 93] that, when ignoring all shadows, the images of a Lambertian scene lie in a three-dimensional linear subspace. This result is used for rendering in [99]. When including attached shadows, the set of images produced by a Lambertian scene can be approximated by a low dimensional linear subspace. This is shown both empirically [8, 50] and analytically [7, 110]. Shadows in a real scene are areas where direct light from a light source cannot reach due to the occlusion by other objects and, thus can provide useful information in recovering the

lighting [69, 106]. In [69], a practical approach is proposed to estimate the illumination distribution from shadows cast on a textured, Lambertian surface. In [106], a graphical model is proposed to estimate the illumination which is modeled as a mixture of von Mises-Fisher distributions and detect the shadows of a scene with textured surfaces from a single image and only coarse 3D information. The sparsity of cast shadows is studied in [112, 103, 120]. The work in [112] shows that, although the set of images produced by a scene with cast shadows can be of high dimension, empirically this dimension does not grow too rapidly. In [103], a sparse representation using a Haar wavelet basis is proposed to recover lighting in images with cast shadows. A method is proposed in [120] to recover an illumination distribution of a scene from image brightness inside shadows cast by an object of known shape. It introduces an adaptive sampling framework for efficient estimation of illumination distribution using a smaller number of sampling directions.

Prior to our work, the idea that a complex lighting environment with cast shadows can be represented with a few point lights is known in previous work [128, 111, 1, 133]. In [111], a signal-processing framework that describes the reflected light field as a convolution of the lighting and BRDF is introduced. This work suggests performing rendering using a combination of spherical harmonics and directional light sources with ray-tracing to check for shadows. Structured importance sampling [1] is introduced to show how to break an environment map into a set of directional lights for image synthesis. It samples an environment map efficiently to render scenes illuminated by distant natural illumination. Lightcuts [133] is a scalable framework for accurately approximating illumination from thousands or millions of point lights using a strongly sublinear algorithm. However, our motivation and algorithms are quite different from theirs.

The studies in [120, 118, 119, 103] are most closely related to our work. These studies propose recovering lighting from cast shadows by a linear combination of basis elements that represent the light. Specifically, in [103] a Haar wavelet basis is used to effectively capture lighting sparsely.

In the field of computer graphics, illumination information is necessary for augmented reality [4], where virtual objects are inserted and seamlessly integrated into a real scene. The recovered lighting can give real world appearance and make images look realistic. Illumination recovery has been of great interest to the graphics community and there has been significant previous research [124, 126, 127, 132, 1, 101, 133]. In [124], a method is proposed to divide the scene into cells and for each cell split the lights into important and unimportant lists with the latter very sparsely sampled. Precomputed radiance functions (PRT) [126] are used to represent low-frequency lighting environments that capture soft shadows, and interreflections. Cluster principal component analysis (CPCA) [127] is used to compress the high dimensional surface signal formed from per-point transfer matrices recorded by PRT. A new data representation and compression technique for precomputed radiance transfer is introduced in [132] and the light transfer functions and light sources are modeled with Spherical Radial Basis Functions (SRBFs). The environment map is approximated in a wavelet basis, and the light transport matrix is sparsely and accurately encoded in the same basis [101]. In [107, 121], a framework for capturing a sparse approximation of the light transport based on the theory of compressive sensing using a small set of illumination patterns is proposed.

There are many other methods to recover illumination distributions from images; though cast shadows are not handled specifically. The complexity of determining lighting

grows dramatically when we must account for cast shadows. A framework is proposed in [100] to accomplish photo-realistic view-dependent image synthesis from a sparse image set and a geometric model. Two methods are presented for recovering the light source position from a single image without the distant illumination assumption [52]. Much more accurate multiple illumination information is extracted from the shading of a sphere [145]. In [91], a framework is proposed to automatically recover an object shape, reflectance properties and light sources from a set of images. A unified framework to estimate both distant and point light sources is proposed in [150]. The number of point light sources and the reflectance property of an object are simultaneously estimated using the EM algorithm [53]. In [73], synthetic 3D objects are inserted into the scene based on the estimated illumination from a single outdoor image. The method relies on a combination of weak cues extracted from different portions of the image such as sky, vertical surfaces, and the ground.

1.3 Particle Filters

In this section, we briefly review particle filters which will be used in tracking in the following chapter. Our proposed ℓ_1 tracker is also based on a particle filter framework.

The particle filter [36, 37, 45, 72, 76] is a Bayesian sequential importance sampling technique for estimating the posterior distribution of hidden state variables, $x_{1:t} = \{x_1, x_2, \cdots, x_t\}$, characterizing a dynamic system from a noisy collection of observations, $z_{1:t} = \{z_1, z_2, \cdots, z_t\}$ in a sequential series. The result is determined by the maxi-

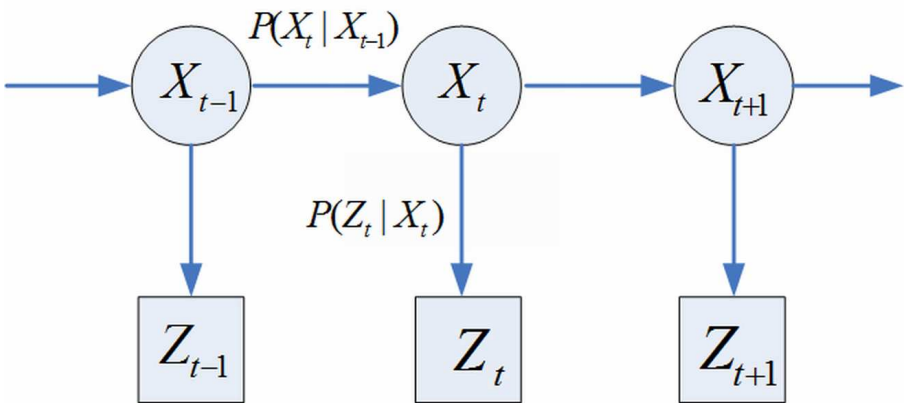


Figure 1.3: Markov chain representation of particle filter.

mum a posteriori (MAP) estimation,

$$\hat{x}_{1:t} = \arg\max_{x_{1:t}} p(x_{1:t}|z_{1:t}) \tag{1.32}$$

Since real time processing is preferred, we adapt a recursive method by maximizing the MAP of $x_t = \text{argmax}_{x_t} p(x_t|z_{1:t})$. This provides a convenient framework for estimating and propagating the posterior probability density function of state variables regardless of the underlying distribution. Two important components of the particle filter are state transition and observation models. The state transition model assumes x_t is a first order Markov process (a graphical model representation is in Figure 1.3) such that

$$x_t = f_t(x_{t-1}, \epsilon_t) \tag{1.33}$$

It defines the temporal correlation of the target states in successive time.

The observations y_ts are conditionally independent provided that x_ts are known. It determines the similarity measurement of a hypothesis against the target model.

$$y_t = g_t(x_t, \zeta_t) \tag{1.34}$$

where both ϵ_t is the system noise and ζ_t is the observation noise. The f_t and g_t are known

functions. If the functions f_t and g_t are linear and both the noise is Gaussian, the Kalman filter [66, 12, 2] finds the exact Bayesian filtering distribution. Otherwise, Kalman filter based methods are a first-order approximation. While for particle filter, the distribution is more accurately modeled by generating enough particles.

Particle filter consists of essentially two steps: prediction and update. Let x_t denote the state variable describing the affine motion parameters of an object at time t. The predicting distribution of x_t given all available observations $z_{1:t-1} = \{z_1, z_2, \cdots, z_{t-1}\}$ up to time $t-1$, denoted by $p(x_t|z_{1:t-1})$, is recursively computed as

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \tag{1.35}$$

At time t, the observation z_t is available and the state vector is updated using the Bayes rule

$$\begin{aligned} p(x_t|z_{1:t}) &= \frac{p(x_t, z_t, z_{1:t-1})}{p(z_t, z_{1:t-1})} \\ &= \frac{p(z_t|x_t, z_{1:t-1})p(x_t|z_{1:t-1})p(z_{1:t-1})}{p(z_t|z_{1:t-1})p(z_{1:t-1})} \\ &= \frac{p(z_t|x_t, z_{1:t-1})p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \end{aligned} \tag{1.36}$$

where $p(z_t|x_t)$ denotes the observation likelihood. $p(z_t|z_{1:t-1})$ is regarded as a normalization factor unrelated to x_t.

In the particle filter, the posterior $p(x_t|z_{1:t})$ is approximated by a finite set of N samples $\{x_t^i\}_{i=1,\cdots,N}$ with importance weights w_t^i. The candidate samples x_t^i are drawn from an importance distribution $q(x_t|x_{1:t-1}, z_{1:t})$ and the weights of the samples are updated as

$$w_t^i = w_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t|x_{1:t-1}, z_{1:t})} \tag{1.37}$$

The samples are resampled to generate a set of equally weighted particles according to their importance weights to avoid degeneracy. In the case of the bootstrap filter $q(x_t|x_{1:t-1}, z_{1:t}) = p(x_t|x_{t-1})$ and the weights become the observation likelihood $p(z_t|x_t)$.

The state transition model characterizes the motion between the frames. In the visual tracking problem, it is ideal to have an exact model to describe the movement of the target. However, simplified models are used to approximate the motion. The motion model can be learned through a training video [60, 102], fitted using a constant velocity model [9, 14, 138, 11, 148], or approximated by introducing an adaptive-velocity model [149]. The observation model uses the measurement extracted from the frame for the inference. The measurement can be appearance model [90, 148, 49, 125, 74], or contours [60].

1.4 Visual Tracking

Visual tracking is a critical task in many computer vision applications such as surveillance, robotics, human computer interaction, vehicle tracking, and medical imaging, etc. Tracking works by finding a region in the current frame that matches a template as closely as possible. The challenges in designing a robust visual tracking algorithm are caused by the presence of noise, occlusion, varying viewpoints, background clutter, and illumination changes [142]. A variety of tracking algorithms have been proposed to overcome these difficulties.

Early works used the sum of squared difference (SSD) as a cost function in the

tracking problem [6]. A gradient descent algorithm was most commonly used to find the minimum [48]. Subsequently, more robust similarity measures have been applied and the mean-shift algorithm or other optimization techniques utilized to find the optimal solution [24]. It iteratively carries out a kernel based search starting at the previous location of the object. The success of the mean shift depends highly on the discriminating power of the histograms that are considered as the objects' probability density function. A mixture model of three components with an online EM algorithm is proposed to model the appearance variation during tracking [62]. However, in [51], an online density-based appearance model uses a varying number of Gaussians to represent the color and color rectangular features at each pixel. The number, mean, covariance, and weight are determined automatically during tracking. The graph-cut algorithm was also used for tracking by computing an illumination invariant optical flow field [43]. The covariance tracker [108] was proposed to successfully track nonrigid objects using a covariance based object description that fuses different types of features and modalities.

Tracking can be considered as an estimation of the state for a time series state space model. The problem is formulated in probabilistic terms. Early works uses a Kalman filter [12] to provide solutions that are optimal for a linear Gaussian model. The particle filter, also known as the sequential Monte Carlo method [36], is one of the most popular approaches. It recursively constructs the posterior probability density function of the state space using Monte Carlo integration. It has been developed in the computer vision community and applied to tracking problems under the name Condensation [61]. In [149], an appearance-adaptive model is incorporated in a particle filter to realize robust visual tracking and classification algorithms. A hierarchical particle filter is used for multiple

object tracking in [139]. In [68], an efficient method for using subspace representation in a particle filter by applying Rao-Blackwellization to integrate out the subspace coefficients in the state vector. A sequential particle swarm optimization framework is proposed for visual tracking inspired by the animal swarm intelligence in evolutionary computing [146]. A multi-view multi-hypothesis approach using particle filter is proposed to segment and track people on a ground plane [71]. In [86], tracking is handled by registering the frames and tracking the moving objects simultaneously within the factorial Hidden Markov Model framework using particle filters. It uses a joint maximum gradient method in [85] to quickly register the successive frames.

Tracking can be considered as finding the minimum distance from the tracked object to the subspace represented by the training data or previous tracking results [10, 56, 84, 114]. In [10], the appearance of the object is represented using an eigenspace. The appearances of objects are represented using affine warps of learned linear subspaces of the image space [56]. In [114], a tracking method is proposed to incrementally learns a low-dimensional subspace representation, efficiently adapting online to changes in the appearance of the target.

Tracking can also be considered as a classification problem and a classifier is trained to distinguish the object from the background [3, 22, 98]. In [3], a feature vector is constructed for every pixel in the reference image and an adaptive ensemble of classifiers is trained to separate pixels that belong to the object from pixels that belong to the background. In [22], a confidence map is built by finding the most discriminative RGB color combination in each frame. A set of discriminant functions each recognizing the object pattern against the background patterns is used to locate the target and the tracker is ro-

bust to view changes when unseen views of the object come into view [98]. A hybrid approach that combines a generative model and a discriminative classifier is used to capture appearance changes and allow reacquisition of an object after total occlusion [144].

A set of auxiliary objects is integrated into the tracking process to provide efficient computation as well as strong verification [140]. Learning also comes to play in visual tracking [135, 5]. Other studies on robust visual tracking handle special topics such as occlusion can be found in [59, 151, 26], etc.

Recently, video based tracking and recognition has gained much attention. In [38], an adaptive framework is proposed for learning human identity by using the motion information along the video sequence, which improves both face tracking and recognition. In [149], a framework is proposed to track and recognize human faces from video simultaneously using a particle filter. The recognition determines the identity of the person which is fixed in the whole video sequence. The parameters to be inferred consist of face motion vector and an identity variable. A form of simultaneous tracking and recognition based on high dimensional nearest neighbor searching is described in [116]. It needs a lot of prototype images taken at different poses to be stored in the image database. In [92], an approach is proposed for object tracking and recognition by combining the feature point matching with optical flow. An adaptive Hidden Markov Models (HMM) is used to perform video-based face recognition [75]. A video based vehicle classification system is proposed in [94]. Detection, tracking, and recognition are integrated to one system and applied to IR video vehicle classification proposed in [87].

Chapter 2

Illumination Recovery from Images with Cast Shadows

2.1 Introduction

Dealing with shadows is an important and challenging problem in the study of illumination effects. Shadows make it difficult to recover illumination from a scene given a single input image. However, as observed in previous studies [112, 103, 120], images with cast shadows can often be sparsely represented, which is attractive since sparsity leads to efficient estimation, dimensionality reduction, and efficient modeling.

In this dissertation, we solve the problem of illumination recovery from a single image with cast shadows and show that the illumination can be well approximated by a combination of low frequency spherical harmonics and a sparse set of directional light sources. As in previous work, we recover lighting using a prior model about the scene that captures geometry and albedos [8, 7, 120, 118, 119, 103, 117]. It is pointed out in [53] that the assumption of known geometry is required by many illumination estimation methods. Illumination recovery with sparse light sources can be very helpful for many applications. For example, by representing the lighting using a sparse set of directional sources, we can save a large amount of time in rendering very complex scenes while maintaining the quality of scene recovery.

We cast the problem of finding a sparse representation of directional sources as an ℓ_1-regularized least squares problem. This is partly motivated by recent advances in

compressive sensing [16, 33]. Compared to ℓ_2 minimization, ℓ_1 minimization tends to find the most significant directional sources and discard the insignificant ones. This is very suitable for our purpose in which we want to select a sparse representation from about one thousand directional sources. The solution to the ℓ_1-regularized least squares problem using the truncated Newton interior-point method is very fast and reliable, which enables our method to be used in many areas, such as lighting design. The proposed method is tested on synthetic and real images in which it outperforms other state-of-the-art approaches in both accuracy and speed.

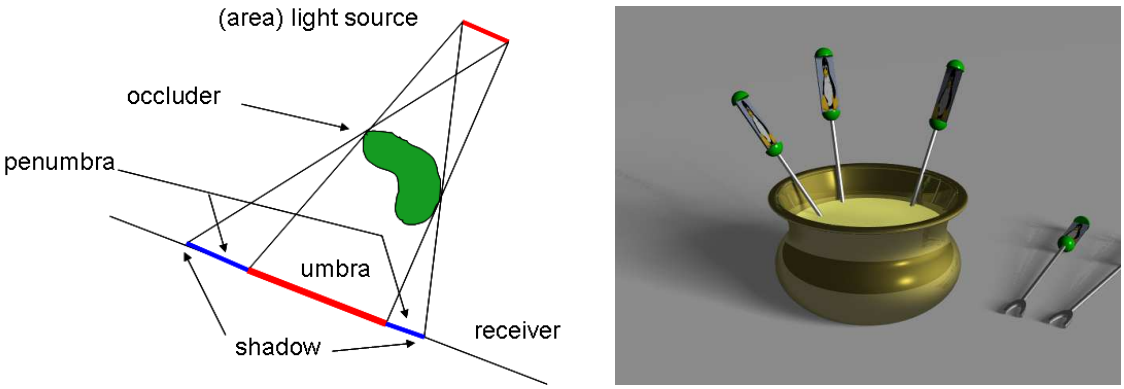
2.2 What are the Shadows?

A shadow is an area where direct light from a light source cannot reach due to obstruction by an object. Shadows in an image provide useful information of the scene: shapes of the objects, relative position of the objects, as well as the characteristics of the light sources. The detection of shadows can aid important computer vision tasks such as image segmentation and object detection.

The first step in the development of efficient tools for recognizing objects with shadows in digital images and image sequences is an understanding of how shadows appear in images and what is peculiar to them. Shadows are crucial for human perception of the 3D world.

The darkest part of a shadow is the *umbra*. A point P of the scene is considered to be in the *umbra* if it is completely blocked by the object causing the shadow, i.e. it doesn't receive any light from the light source. If the light is partially blocked, hence P

can view a part of the light source, it is in the *penumbra*. The union of the umbra and the penumbra is the shadow, the set of points for which at least one point of the light source is occluded. Objects that hide a point from the light source are called occluders. Figure 2.1 (a) shows the generation of umbra and penumbra. Area light sources emit light to the receiver and some lights are blocked by the occluder. The blue part in the penumbra on the receiver can only see part of the light source, while the red part in the umbra cannot see anything from the light source. The umbra and penumbra structure is clearly visible in figure 2.1 (b).



(a) (b)

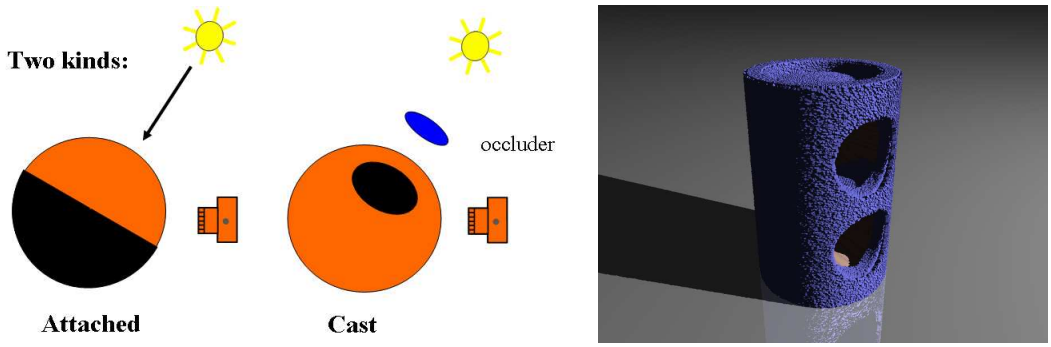
Figure 2.1: Umbra and penumbra generation. Area light sources, as the line source in (a), generate penumbra where the light is only partially obstructed by the shadow casting object. The umbra and penumbra structure is clearly visible in (b).

Shadows are categorized as two types:

- attached shadows or self shadows, that occur when a surface faces away from a light source.
- cast shadows, that occur when an intervening part of an object blocks the light from

reaching a different part of the surface.

For convex objects, only attached shadows occur. Attached shadows are easily modeled, since they depend only upon the local geometry of the surface, that is, the normal direction of the point on the surface. On the other hand, cast shadows are caused when an entirely different region of the surface intersects the path from the light source to the point. Since they are dependent on the global geometry of the surface, cast shadows are more complex to model. Figure 2.2 (a) illustrates the generation of attached and cast shadows. An example of attached and cast shadows in real life is given in figure 2.2 (b). The left side of the cat condo, which faces away from the light source, is in the attached shadow. The cat condo casts a hard shadow on the background.

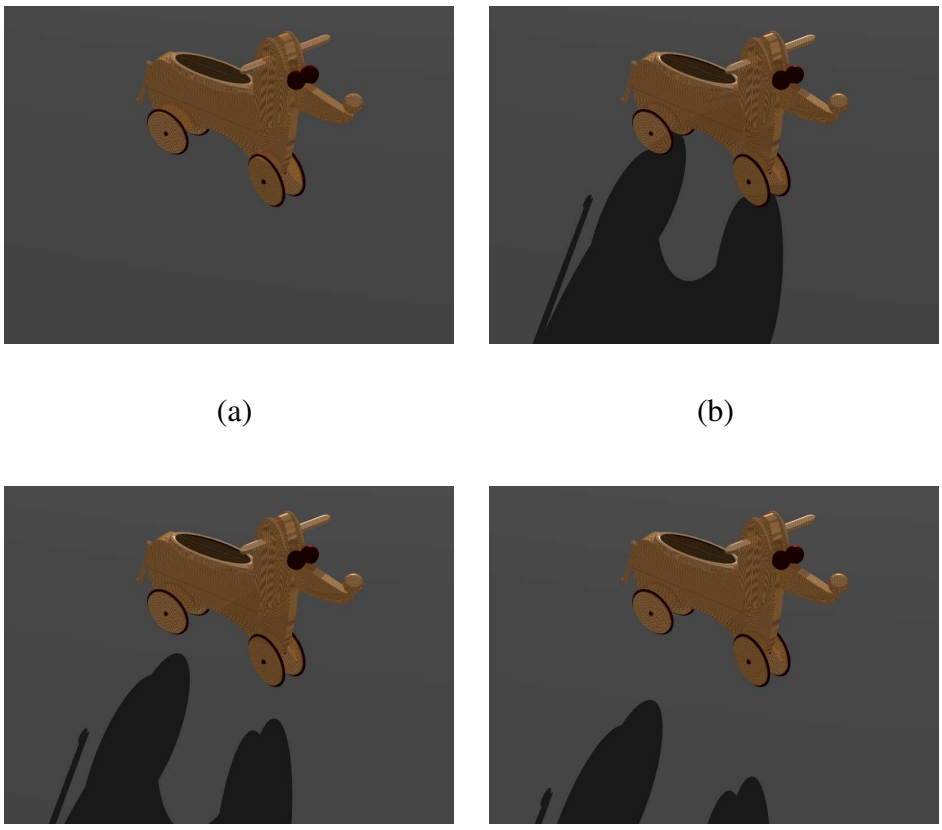


(a) (b)

Figure 2.2: Examples of attached and cast shadow. The generation of attached and cast shadow is illustrated in (a). (b) gives an example of attached and cast shadows in real life. The left side of the cat condo, which faces away from the light source, is in the attached shadow. The cat condo casts a hard shadow on the background.

Shadows provide a strong source of information about the shape of surfaces, which we now illustrate with a concrete example.

Shadows help us to understand relative object position and size in a scene. In figure 2.3 (a), we are unable to determine the position of an object in space without a cast shadow, whereas on the other three images we understand that it is more and more distant from the ground.



(a) (b)

(c) (d)

Figure 2.3: Shadows provide information about the relative positions of objects. We cannot determine the position of the wooden dog from shadowless image (a), whereas on the other three images we understand that it is more and more distant from the ground.

Shadows can also provide information about the shape and geometry of a complex occluder. From the shadow cast on the ground in figure 2.4 (a), we can see the occluder is a man on a racing horse. We can also tell there is a pin sticking in the ground from 2.4

(b).

(a) (b)

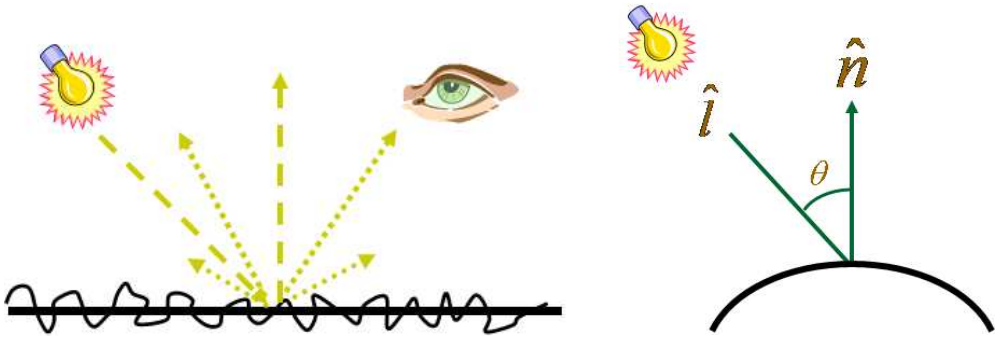
Figure 2.4: Shadows provide information about the shape and geometry of the occluder.

2.3 Lambertian Model

We assume that a surface exhibits Lambertian reflectance, which states that light falling on it is reflected equally in all directions and the brightness of the surface is the same to an observer regardless of the observer's angle of view. In figure 2.5 (a), the light falling on the Lambertian surface is reflected equally in all directions and appears the same to the viewer from any viewing direction. The bidirectional reflection distribution function(BRDF) for a Lambertian surface is known to be a constant. Under the Lambertian law, the intensity of a point p is calculated by taking the dot product of the surface normal $n(p)$ and lighting direction l with multiplication by albedo ρ and light intensity $i(l)$. (see figure 2.5 (b))

$$I(p) = \rho i(l) n(p) \cdot l \tag{2.1}$$

If the image of a Lambertian object has no shadows, the set of all images under all lighting conditions is a 3-dimensional space. For any lighting direction l, it can be



(a) (b)

Figure 2.5: Lambertian surface and law. In (a), the light falling on the Lambertian surface is reflected equally in all directions and appears the same to the viewer from any viewing direction. According to the Lambertian law, the intensity of a point is the dot product of the surface normal and lighting direction as in (b).

represented by 3 non-coplanar basis lighting directions $\{l_1, l_2, l_3\}$. Given that n is the surface normal of a surface point, the image brightness value of the point is

$$I(p) = \rho i(l) n(p) \cdot l = \rho i(l) n \cdot (a_1 l_1 + a_2 l_2 + a_3 l_3) = \rho i(l)(a_1 n(p) \cdot l_1 + a_2 n(p) \cdot l_2 + a_3 n(p) \cdot l_3) \tag{2.2}$$

The brightness value of a surface point is a linear combination of the brightness value of the same point under 3 fixed, independent illuminations. Thus the images of a Lambertian scene without shadows lie in a three-dimensional subspace [122, 93].

2.4 Spherical Harmonic Analysis

Problems arise when the point is in the attached shadow. The angle between the surface normal and the lighting direction is obtuse ($n(p) \cdot l < 0$), which results in the space for the set of images under all lighting conditions no longer being 3-dimensional.

In [7], it is shown that it is very close to a 9D linear subspace using spherical harmonic analysis. This is analogous to Fourier analysis, but on the surface of a sphere. With a spherical harmonic representation, low frequency light means light whose intensity varies slowly as a function of direction. Image formation is analogous to the convolution of the lighting function with a half cosine function. The reflectance function acts as a low-pass filter with 99.2 percent of its energy in the first nine components.

In the following, we briefly review the spherical harmonic analysis for attached shadows. We denote the lighting direction and surface normal using unit vectors u_l and v_r, respectively. According to Lambert's law, if a light ray of intensity l coming from direction u_l reaches a surface point with albedo ρ and normal direction v_r, then the intensity, i, reflected by the point due to this light is given by

$$i = \rho l(u_l) max(u_l \cdot v_r, 0) \tag{2.3}$$

If we fix the lighting and ignore ρ for now, we can write

$$r(v_r) = \int_{S^2} k(u_l \cdot v_r) l(u_l) du_l \tag{2.4}$$

where $k(u \cdot v) = max(u \cdot v, 0)$. The reflected light on a point is a function of surface normal alone. Intuitively, it can be regarded as a convolution of k and l.

The surface spherical harmonics are a set of functions that form an orthonormal basis for the set of all functions on the surface of the sphere. Any piecewise continuous function f on the surface of the sphere can be written as a linear combination of an infinite series of harmonics. Specifically, for any f,

$$f(u) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} f_{nm} Y_{nm}(u) \tag{2.5}$$

where f_{nm} is a scalar value, computed as:

$$f_{nm} = \int_{S^2} f(u) Y^*_{nm}(u) du \tag{2.6}$$

The first nine spherical harmonics, which are a function of space coordinates (x, y, z) are:

$$\begin{aligned}
Y_{00} &= \tfrac{1}{\sqrt{4\pi}} & Y_{10} &= \sqrt{\tfrac{3}{4\pi}} z \\
Y^e_{11} &= \sqrt{\tfrac{3}{4\pi}} x & Y^o_{11} &= \sqrt{\tfrac{3}{4\pi}} y \\
Y_{20} &= \tfrac{1}{2}\sqrt{\tfrac{5}{4\pi}}(3z^2 - 1) & Y^e_{21} &= 3\sqrt{\tfrac{5}{12\pi}} xz \\
Y^o_{21} &= 3\sqrt{\tfrac{5}{12\pi}} yz & Y^e_{22} &= \tfrac{3}{2}\sqrt{\tfrac{5}{12\pi}}(x^2 - y^2) \\
Y^o_{22} &= 3\sqrt{\tfrac{5}{12\pi}} xy
\end{aligned} \tag{2.7}$$

where the superscripts e and o denote the even and odd components of the harmonics, respectively. $Y_{nm} = Y^e_{n|m|} \pm i Y^o_{n|m|}$. Figure 2.6 shows the first nine spherical harmonics.

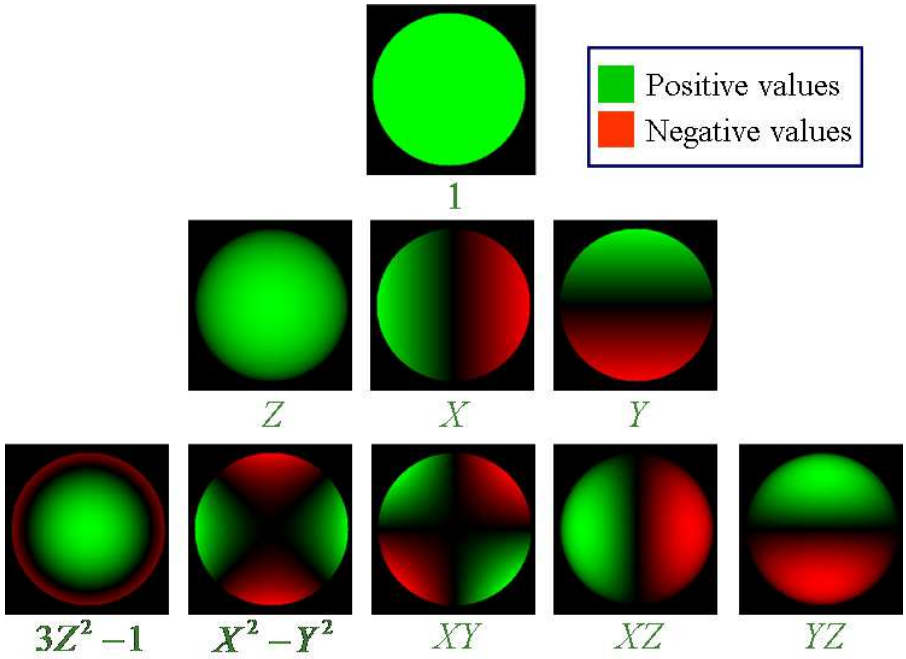


Figure 2.6: The first nine spherical harmonics.

Both the lighting function l, and Lambertian kernel k, can be written as sums of

spherical harmonics. Denoted by

$$l = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} l_{nm} Y_{nm} \tag{2.8}$$

the harmonic expansion of l, and by

$$k(u) = \sum_{n=0}^{\infty} k_n Y_{n0} \tag{2.9}$$

Note that since $k(u)$ is circular symmetric about the north pole

$$\int_{S^2} k(u) Y_{nm}^*(u) du = 0, m \neq 0 \tag{2.10}$$

According to the Funk-Hecke theorem, the harmonic expansion of the reflectance function r can be written as:

$$r = k * l = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} (\alpha_n l_{nm}) Y_{nm} \tag{2.11}$$

where $\alpha = \sqrt{\frac{4\pi}{2n+1}} k_n$.

The first few coefficients of Lambertian kernel are

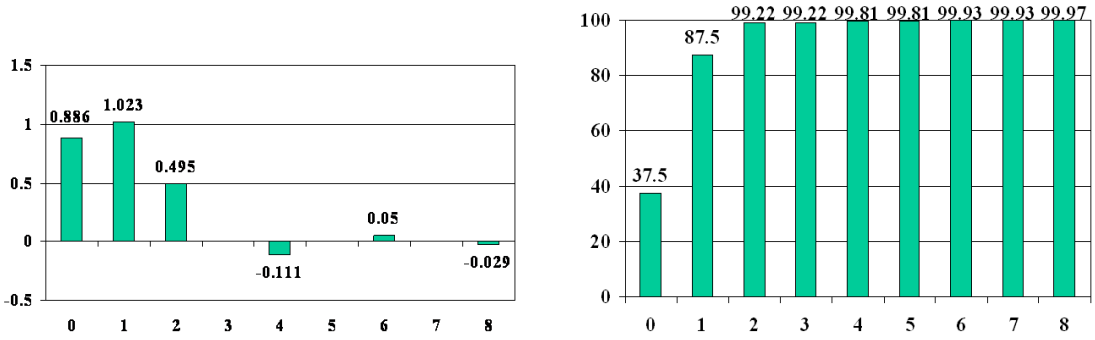
$$\begin{aligned} k_0 &= \tfrac{\sqrt{\pi}}{2} \approx 0.8862 & k_1 &= \sqrt{\tfrac{\pi}{3}} \approx 1.0233 \\ k_2 &= \tfrac{\sqrt{5\pi}}{8} \approx 0.4954 & k_4 &= -\tfrac{\sqrt{\pi}}{16} \approx -0.1108 \\ k_6 &= \tfrac{\sqrt{13\pi}}{128} \approx 0.0499 & k_8 &= -\tfrac{\sqrt{17\pi}}{256} \approx -0.0285 \end{aligned} \tag{2.12}$$

Figure 2.7 is a graph representation of the first 9 coefficients and the cumulative energy of the Lambertian kernel. Because the Lambertian kernel k acts as a low-pass filter, the high-frequency components of the lighting have little effect on the reflectance function. We achieve a low-dimensional approximation to the reflectance function by truncating the sum in (2.11).

$$r = k * l = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} (\alpha_n l_{nm}) Y_{nm} \approx \sum_{n=0}^{N} \sum_{m=-n}^{n} (\alpha_n l_{nm}) Y_{nm} = \sum_{n=0}^{N} \sum_{m=-n}^{n} l_{nm} r_{nm} \tag{2.13}$$

where r_{nm} is given by

$$r_{nm} = k * Y_{nm} = \alpha_n Y_{nm} \tag{2.14}$$



(a) (b)

Figure 2.7: A graph representation of the first 9 coefficients and the cumulative energy of the Lambertian kernel.

Using this analysis, we can efficiently represent the set of images of objects seen under varying illumination. Let p_i denote the ith point on the surface of the object. Let n_i and ρ_i denote the surface normal and albedo of p_i, respectively. Then, the image I_i of p_i is:

$$I_i = \rho_i r(n_i) = \rho_i \sum_{n=0}^{\infty} \sum_{m=-n}^{n} l_{nm} r_{nm} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} l_{nm} b_{nm}(p_i) \tag{2.15}$$

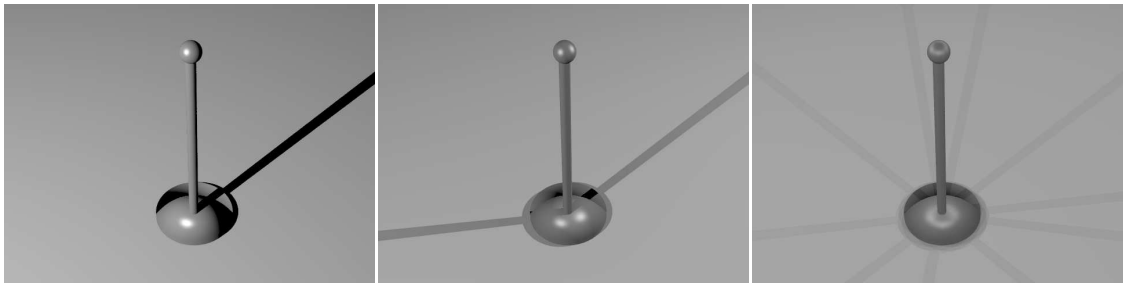
where $b_{nm}(p_i) = \rho_i r_{nm}(n_i)$ is the harmonic image. We can see any image is a linear combination of harmonic images.

We can recognize objects by comparing a new query image to the linear subspace of images that correspond to each model in turn. Given an image I, we seek the distance from I to the space spanned by the basis images. Let B denote the basis images. Then we seek a vector a that minimizes $||Ba - I||$. Every column of B contains one harmonic image b_{mn}. This shows very good results for face recognition.

2.5 An Example

Although the set of images produced by a scene with cast shadows can be of high dimension, the number of directional lights needed to approximate the lighting is highly compressible and the perceptual loss from the image constructed by the recovered lighting is hardly noticeable. To strengthen our intuitions, we consider a very simple example of a scene consisting of a flat playground with an infinitely thin flag pole. We view the scene from directly above, so that the playground is visible, but the flag pole appears only as a negligible point. Suppose the scene is illuminated by an arbitrary set of directional lights of equal intensity that each has an elevation of 45 degrees. In this case, the intensity of the lighting can be described as a one-dimensional function of azimuth. A single directional light illuminates the playground to constant intensity except for a thin, black shadow on it. The entire set of lights can cause shadows in multiple directions. None of these shadows overlap, because the pole is infinitely thin.

Now consider the linear subspace spanned by the images that this scene can produce. We first consider the set of images that are each produced by a single directional source. All images are nonnegative, linear combinations of these. We represent each image as a vector. By symmetry, the mean of these images will be the constant image produced in the absence of cast shadows. Subtracting the mean, each image is near zero, except for a large negative component at the shadow. All these images have equal magnitude, and are orthogonal to each other. Therefore, they span an infinite-dimensional space, and Principal Component Analysis (PCA) will produce an infinite number of equally significant components. A finite-dimensional linear subspace cannot capture any significant



(a) (b) (c)

Figure 2.8: A flagpole rendered with one directional source (a), two directional sources (b), and ten directional sources (c). The shadows are lighter as the number of directional sources increases.

fraction of the effects of cast shadows.

But, let's look at the images of this scene differently. A single directional source produces a single, black shadow (Figure 2.8(a)). Two sources produce two shadows (Figure 2.8(b)), but each shadow has half the intensity of the rest of the playground, because each shadow is lit by one of the lights. The more lights (e.g., Figure 2.8(c)) we have the more shadows we have, but the lighter these shadows are. Therefore, while a sparse set of lights can produce strong cast shadows, many lights tend to wash out the effects of shadowing.

Now, suppose we approximate any possible image using one image of constant intensity, and a small number of images that are each produced by a directional source. If the actual image is produced by a small number of directional sources, we can represent its shadows exactly. If the image is produced by a large number of directional sources, we cannot represent the shadows well with a few sources, but we do not need to, because they have only a small effect and the image is approximately constant.

2.6 Modeling Images with Cast Shadows

We now model cast shadows in detail. We do not consider specular reflections, and in fact there is no reason to believe that sparse lighting can approximate the effects of full environment map lighting when there are significant specular reflections (instead of our playground example, imagine images of a mirrored ball. Directional sources produce bright spots which do not get washed out as we add more directional sources). We also do not consider the effects of saturated pixels. We assume the geometry of the scene is given, so we can render directional source images from it.

A scene is illuminated by light from all directions. Therefore, an image $I \in \mathbb{R}^d$ (we stack image columns to form a 1D vector) of a given scene has the following representation

$$I = \int_{\mathcal{S}} x(\theta) I_{dir}(\theta) d\theta \ , \quad x(\theta) \geq 0 \ , \tag{2.16}$$

where $\mathcal{S}$ is the unit sphere that contains all possible light directions, $I_{dir}(\theta)$ is the image generated by a directional light source with angle $\theta \in \mathcal{S}$, and $x(\theta)$ is the weight (or amount) of image $I_{dir}(\theta)$.

For practical reasons, integration over the continuous space $\mathcal{S}$ is replaced by a superposition over a large discrete set of lighting directions, say $\{\theta_k\}_{k=1}^N$ with a large N. Denote the image generated by light from direction θ_k as $I_k = I_{dir}(\theta_k)$ and $x_k = x(\theta_k)$; we approximate with the discrete version of (2.16),

$$I = \sum_{k=1}^{N} x_k I_k \ , \quad x_k \geq 0. \tag{2.17}$$

It is known that in the absence of cast shadows, this lighting can be approximated using low frequency spherical harmonics [7, 110]. We use a nine-dimensional spherical

harmonic subspace generated by rendering images of the scene, including their cast shadows, using lighting that consists of zero, first, and second order spherical harmonics. We will therefore divide the effects of these directional sources into low- and high-frequency components. We can then capture the low-frequency components exactly using a spherical harmonic basis. We will then approximate the high frequency components of the lighting using a sparse set of components that each represent the high frequency part of a single directional source.

We project the directional source image I_k onto the spherical harmonic subspace and it can be written as the sum of the projection image $\hat{I}_k$ and residual image $\tilde{I}_k$. Then Equation (2.17) can be written as:

$$I = \sum_{k=1}^{N} x_k(\hat{I}_k + \tilde{I}_k) \ , \quad x_k \geq 0. \tag{2.18}$$

We separate the low frequency component $\hat{I}_k$ from high frequency component $\tilde{I}_k$ and Equation 2.18 becomes:

$$I = \sum_{k=1}^{N} x_k\hat{I}_k + \sum_{k=1}^{N} x_k\tilde{I}_k \ , \quad x_k \geq 0. \tag{2.19}$$

We know that the low frequency component $\sum_{k=1}^{\infty} x_k\hat{I}_k$ lies in a low dimensional subspace and can be approximated using $\hat{I}$ by simply projecting I to the spherical harmonic subspace. Equation (2.19) can be written as:

$$I = \hat{I} + \sum_{k=1}^{N} x_k\tilde{I}_k \ , \quad x_k \geq 0. \tag{2.20}$$

$\hat{I}$ is simply the component of the image due to low-frequency lighting, where we solve for this component exactly using the method of [7]. We then approximate the high frequency components of the lighting using a sparse set of values for x_k. Note that these

components will be reflected only in the cast shadows of the scene, and we expect that when these cast shadows are strong, a sparse approximation will be accurate.

Our problem is now reduced to finding a certain number of x_k's that best approximate the residual image $\tilde{I} = I - \hat{I}$. It can be addressed as a least squares (LS) problem with nonnegativity constraints:

$$\arg\min_{\mathbf{x}} ||A\mathbf{x} - \tilde{I}||^2, \quad x_k \geq 0 \,, \tag{2.21}$$

where $A = [\tilde{I}_1 \tilde{I}_2 \cdots \tilde{I}_N]$ and $\mathbf{x} = (x_1, ..., x_N)^\top$. To avoid ambiguity, we assume all the residual directional source images $\tilde{I}_k$ are normalized, i.e., $||I_k||_2 = 1$.

The size of the image can be very large, which corresponds to a large linear system $Ax = \tilde{I}$. To reduce the dimensionality and speed up the computation, we apply PCA to the image set A. The standard PCA yields a projection matrix $W \in \mathbb{R}^{m \times d}$ that consists of the m most important principal components of A.

Applying W to equation 2.21 yields:

$$\begin{aligned} & \arg\min_{\mathbf{x}} ||W(A\mathbf{x} - \mu) - W(\tilde{I} - \mu)||^2 \\ = \; & \arg\min_{\mathbf{x}} ||WA\mathbf{x} - W\tilde{I}||^2, \quad x_k \geq 0 \,, \end{aligned} \tag{2.22}$$

where μ is the mean vector of columns of A.

The dimension m is typically chosen to be much smaller than d. In this case, the system (2.22) is underdetermined in the unknown $\mathbf{x}$ and simple least squares regression leads to over-fitting.

2.7 ℓ_1-Regularized Least Squares

A standard technique to prevent over-fitting is ℓ_2 or Tikhonov regularization [96], which can be written as

$$\arg\min_{\mathbf{x}} ||WA\mathbf{x} - W\tilde{I}||^2 + \lambda||\mathbf{x}||_2, \quad x_k \geq 0. \tag{2.23}$$

where $||\mathbf{x}||_2 = (\sum_{k=1}^{N} x_k^2)^{1/2}$ denotes the ℓ_2 norm of $\mathbf{x}$ and $\lambda > 0$ is the regularization parameter.

We are concerned with the problem of low-complexity recovery of the unknown vector $\mathbf{x}$. Therefore, we exploit the compressibility in the transform domain by solving the problem as the ℓ_1-regularized least squares problem. We substitute a sum of absolute values for the sum of squares used in Tikhonov regularization:

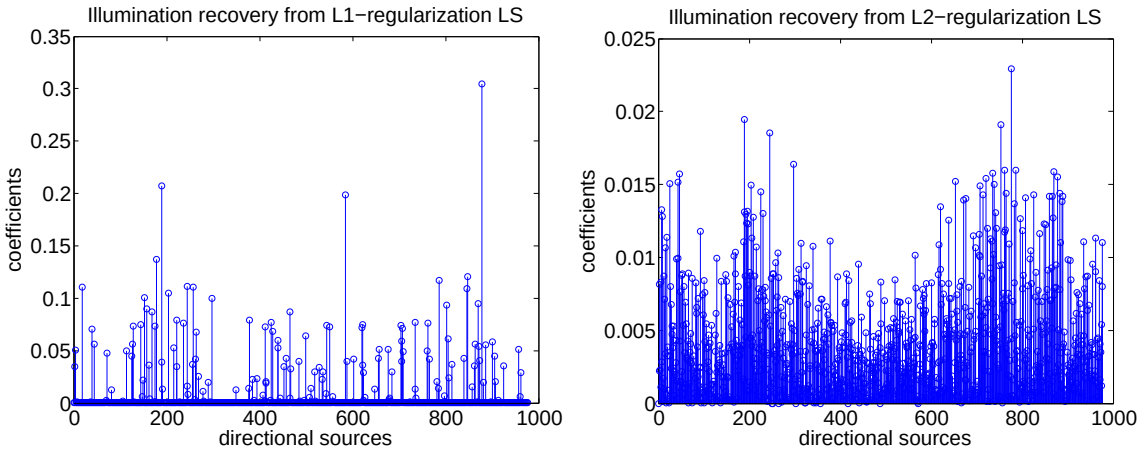
$$\arg\min_{\mathbf{x}} ||WA\mathbf{x} - W\tilde{I}||^2 + \lambda||\mathbf{x}||_1, \quad x_k \geq 0. \tag{2.24}$$

where $||\mathbf{x}||_1 = \sum_{k=1}^{N} |x_k|$ denotes the ℓ_1 norm of $\mathbf{x}$ and $\lambda > 0$ is the regularization parameter. This problem always has a solution, though not necessarily unique. ℓ_1-regularized least squares (LS) typically yields a sparse vector $\mathbf{x}$, which has relatively few nonzero coefficients. In contrast, the solution to the Tikhonov regularization problem generally has all coefficients nonzero.

Since $\mathbf{x}$ is non-negative, the problem (2.24) can be reformulated as

$$\arg\min_{\mathbf{x}} ||WA\mathbf{x} - W\tilde{I}||^2 + \lambda \sum_{k=1}^{N} x_k, \quad x_k \geq 0. \tag{2.25}$$

Figure 2.9 shows the recovered coefficients $\mathbf{x}$ using ℓ_1-regularized LS and ℓ_2-regularized LS algorithms respectively for the synthetic image rendered with the light probe in Figure



(a) (b)

Figure 2.9: The recovered coefficients x from ℓ_1-Regularized LS (a) and ℓ_2-Regularized LS (b).

2.10 (a). The query image is approximated using N=977 directional source images. The parameter λ's are tuned such that the two recoveries have similar errors. The results show that ℓ_1 regularization gives a much sparser representation, which fits our expectation.

Algorithm 2 summarizes the whole illumination recovery procedure. Our implementation solves the ℓ_1-regularized least squares problem via an interior-point method based on [70]. The method uses the preconditioned conjugate gradients (PCG) algorithm to compute the search direction and the run time is determined by the product of the total number of PCG steps required over all iterations and the cost of a PCG step. We use the code from [27] for the minimization task in (2.25).

Algorithm 2 Sparse representation for inverse lighting

1: Obtain N directional source images by rendering the scene with N directional light sources uniformly sampled from the upper hemisphere (object is put on a plane and there is no light coming from beneath).

2: Create the first nine spherical harmonic images by integrating the N directional source images.

3: Project each directional source image I_k to the 9D spherical harmonic subspace and obtain the corresponding residual directional source image $\tilde{I}_k$.

4: Normalize $\tilde{I}_k$ such that $||\tilde{I}_k||_2 = 1$.

5: Generate matrix $A = [\tilde{I}_1 \tilde{I}_2 \cdots \tilde{I}_N]$.

6: Apply Principal Component Analysis to matrix A and obtain the projection matrix W by stacking the m most important principal components of A.

7: Project the query image I to the spherical harmonic subspace and obtain the residual image $\tilde{I}$.

8: Solve the ℓ_1-regularized least squares problem with nonnegativity constraints (2.25).

9: Render the scene with the spherical harmonic lighting plus the recovered sparse set of directional light sources.

2.8 Experiments

In this section, we describe our experiments for illumination recovery on both synthetic and real data, in comparison with four other previous approaches.

2.8.1 Experimental Setup

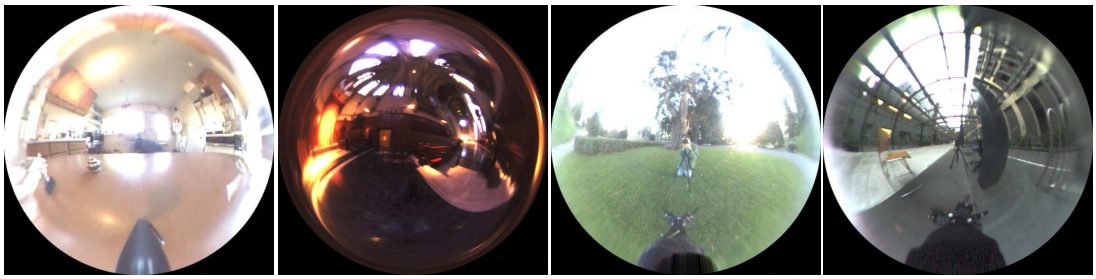
2.8.1.1 Data

Both synthetic and real datasets are used in our experiments to validate the performance of our proposed method.

There are two synthetic scenes used in our experiments. One is composed of a coffee cup and a spoon, with a plate underneath them (see Figure 2.14 and 2.16). The teacup and spoon will cast shadows on the plate when the light comes from certain directions. The other scene consists of one table and four chairs (see Figure 2.15 and 2.17). The legs of table and chairs are thin and will cast shadows on the ground which poses challenges for the illumination recovery. Four synthetic images for each scene are obtained by rendering the scene with environment maps (namely *kitchen*, *grace*, *campus*, and *building*, see Figure 2.10) provided by [31]. We considered a scene where the objects were placed on an infinite plane, so only lights coming from the upper hemisphere are taken into account.

For the real objects, we built CAD models of three objects (namely *chair1*, *chair2*, and *couch*, see Figures 2.19 and 2.20) and printed them with a 3D printer.

The 3D printer we use is manufactured by Z Corporation with the model ZPrinter 310 Plus. The images of the printer ZPrinter 310 Plus are show in Figure 2.11. We briefly explain how a 3D object is created from a 3D printer. First, a 3D CAD file is imported into the system software. The software slices the file into thin cross-sectional slices, which are fed to the 3D printer. Second, the printer creates the model one layer at a time by spreading a layer of powder and inkjet printing a binder in the cross-section of the



(a) (b) (c) (d)

Figure 2.10: Light probes [31] used to generate our synthetic dataset: (a) kitchen, (b) grace, (c) campus, and (d) building. The light probes are sphere maps and shown in low-dynamic range for display purposes.

part. Finally, the process is repeated until every layer is printed and the part is complete and ready to be removed.

We took pictures of the objects and rendered directional source images using PovRay. Figure 2.12 shows the objects we used for the experiments. We name them chair1, couch, and chair2 from left to right. The only difference between chair1 and chair2 is the number of backrest bars. These objects are placed under natural indoor illumination and images are taken by a Canon EOS Digital Rebel XT camera.

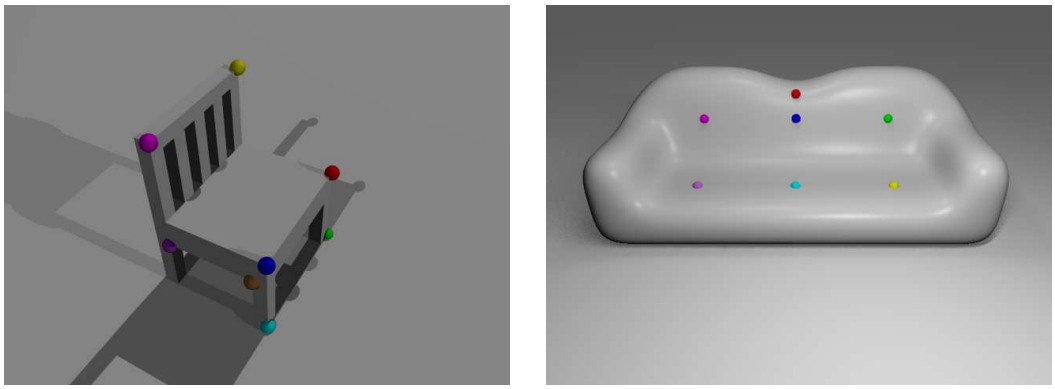
One of our experiments involves recovering lighting from one object (chair1), and using it to render a model of a second object (chair2) [103]. For this reason, we take pictures of chair1 and chair2 in exactly the same illumination environment.

2.8.1.2 Registration

After we take pictures of the real objects, we need to register the object in the picture to the 3D model. We select the feature points and match them from the picture to the 3D

Figure 2.11: Images of the 3D printer.

Figure 2.12: Real objects we used for the experiments. We name them chair1, couch, and chair2 from left to right.



(a) (b)

Figure 2.13: Images of the 3D model with feature points hightlighted using colored balls.

model. Figure 2.13 shows the 3D model with the feature points on it. Figure 2.13 (a) is the chair1 model. The eight corner points are selected as feature points and highlighted with colored balls. Figure 2.13 (b) is the couch model. There is no apparent corner points on the model. We add seven small balls with center on the couch whose center are used as feature points for the registration.

To do the registration, the object is first rotated and translated to the camera coordinate system and then projected onto the image plane. We use a simplified pinhole camera model to do the registration. There will be six parameters for the coordinate transformation and one for the camera.

The objects are registered to the images by minimizing the distance between the feature points on the image and the corresponding feature points from the 3D model. We adapt a simple pinhole camera model [54]. The 3D point position in world coordinates $[X, Y, Z, 1]^T$ expressed in Homogeneous coordinates is projectively mapped to the pixel

coordinates $[u, v, 1]^T$ after translation and rotation through the pinhole camera model.

$$z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = Pz\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K[Rt]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K[R_x * R_y * R_z t]\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.26}$$

K is the intrinsic matrix and contains only one parameter: focal length f.

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.27}$$

R is the rotation matrix which encompass three rotation matrix along three axis. t is the translation vector.

2.8.1.3 Methods for Comparison

We compare our proposed algorithms with Spherical Harmonics [7, 110], Non-Negative Least squares (NNL) [8], Semidefinite Programming (SDP) [117], and Haar wavelets [103] algorithms . To make this algorithm comparable, we use 100 directional sources to represent the lighting which is obtained from thousands of possible directional sources. The reason that we choose 100 sources is because by examining the coefficients for all the experiments, we find that the coefficients become zero or very small after 100 sources. The λ in Equation (2.25) is set to 0.01 for all the experiments.

In the following, we briefly review and analyze the four algorithms that are used for comparison.

1. **Spherical Harmonics**: From Equation 2.15, we can see the image without cast shadows is a linear combination of harmonic images. In [7, 110], it is shown that the image is well approximated using the first nine harmonics images.

2. Non-Negative Linear: In [8], it has shown that the set of images of an object produced by nonnegative lighting is a convex cone in the space of all possible images. Given an image I, we attempt to minimize $||Ha - I||$ subject to $a >= 0$ where H is the matrix whose columns are directional source images. If we densely sample the illumination distribution, it makes the solution exceedingly expensive in terms of processing time and storage requirements because of the high dimensionality of the matrix H formed by point source images.

3. **Semidefinite Programming (SDP)**: In [117], semidefinite programming is applied to perform a constrained optimization to quickly and accurately solve for the non-negative linear combination of spherical harmonics. It has been successfully applied for specular object recognition on both synthetic and real data by better separating the correct and incorrect models. This work is the first to show that SDP can also be used to handle shadows. Their SDP algorithm is summarized as follows:

   (a) First, spherical harmonic images are obtained by rendering the 3D objects with each individual harmonic lighting.

   (b) Second, these harmonic images are vectorized and stacked as columns of a matrix M. The resulting image is described as Ma which is the product of spherical harmonic images M and coefficient vector a.

(c) Third, given the query image $r = I + noise$, a is found by minimizing $||Ma - r||$ subject to $T_L(a) \geq 0$. where

$$\mathbf{T_L(a)} = \begin{bmatrix} a_0 & a_1 & \cdots & a_n \\ a_{-1} & a_0 & \ddots & \\ \vdots & \ddots & \ddots & a_1 \\ a_{-n} & & a_{-1} & a_0 \end{bmatrix}$$

(d) Fourth, compute the QR decomposition of the matrix M. The problem is reduced to: solve the size $(L+1)^2$ problem: $\min_a ||Ra - Q^T r||^2$ subject to $T_L(a) \geq 0$. These kind of problems are called semidefinite programming (SDP) problems.

(e) Finally, the problem is solved as: $\min_a q$ subject to

$$1 + q \geq \begin{bmatrix} 1 - q \\ Ra - Q^T r \end{bmatrix}$$

and $T_L(a) \geq 0$. It's solved in MATLAB using SDPT3 and YALMIP packages.

SDP is designed to approximate high frequency signals that cannot be captured by the 9D spherical harmonics. It works well on specular objects such as a shiny rubber ball and a ceramic shaker using harmonics up to 10th order. The total harmonics used in SDP is $(10+1)^2 = 121$. Since images with cast shadows generally have a lot of high frequency signals, it still misses a certain amount of information which is contained in higher order harmonics.

4. **Haar Wavelets**: In [103], spherical harmonics is compared to Haar wavelets as a basis for recovery from shadows. The Haar wavelets approach is similar since

Haar wavelets also form an orthonormal basis. The advantages pointed out are the compact supports with various sizes which allow different resolutions in different regions.

The illumination distribution is mapped to a cube and two-dimensional Haar wavelet basis elements are used in each face of the cube. Similar to the spherical harmonics (2.15), the illumination distribution is represented by a linear combination of the basis functions $\Phi_i(\theta, \phi)$ and $\Psi_{ijkl}(\theta, \phi)$

$$L(\theta, \phi) = \sum_i (c_i \Phi_i(\theta, \phi) + \sum_{j,k,l} d_{i,j,k,l} \Psi_{ijkl}(\theta, \phi)) \tag{2.28}$$

where c_i and $d_{i,j,k,l}$ are coefficients of the corresponding basis functions. The coefficients are computed using a constrained least squares estimation which constrains the resulting distribution to be position.

2.8.1.4 Evaluation Criterion

Accuracy. To evaluate the accuracy of different algorithms, we use the Root-Mean-Square (RMS) errors of pixel values, which is also used in [103]. Specifically, for an input image $I \in \mathbb{R}^d$ and its recovery $\hat{I} \in \mathbb{R}^d$, the RMS between them is defined as $r(I, \hat{I}) = ||I - \hat{I}||_2$.

Run Time. We divide the run time for illumination recovery into three parts: (1) preprocessing time, (2) time for solving the lighting recovery algorithm (e.g., solving the ℓ_1-regularized LS, SDP), and (3) rendering the scene with recovered lighting. First, part (1) can be done off-line and is actually similar for all methods. In fact, preprocessing time is dominated by the time for generating images using different directional light sources

(via PovRay for all experiments). These images are pre-computed off-line, and are actually used in all methods.[1] Second, part (3) is usually much faster than the other two parts and therefore can be ignored. For the above reasons, in the paper we focus only on the part (2), which measures the time efficiency of different illumination recovery approaches.

All the algorithms were run in MATLAB 7.4.0. The computer used was a Intel Core Duo at 1.73GHz with 2.0GB RAM laptop.

2.8.2 Experiments with Synthetic Data

In this section, we deal first with synthetic data, showing that illumination can be accurately recovered by using our proposed method. Comparison between our proposed method and other methods is made in terms of accuracy and speed.

Using the POV-Ray ray tracer we generate directional images, each using a single directional light source. We obtain directions by uniformly sampling the upper hemisphere. Using these images, we numerically integrate to compute nine images of the scene, each with lighting consisting of a single spherical harmonic.

Figure 2.14 and Figure 2.15 show the first nine harmonic images created from more than three thousand directional source images derived from a 3D model of one teacup, and one table with four chairs. The top row contains the zeroth harmonic (left) and the three first order harmonic images (right). The second row shows the images derived from the second harmonics. Image values are scaled to the full range of the intensity.

For the evaluation, we used synthetic images rendered with four environment maps provided by [31], from high dynamic range light probe images, and recovered illumina-

[1] The spherical harmonics and Haar wavelets also need these images for basis image estimation.

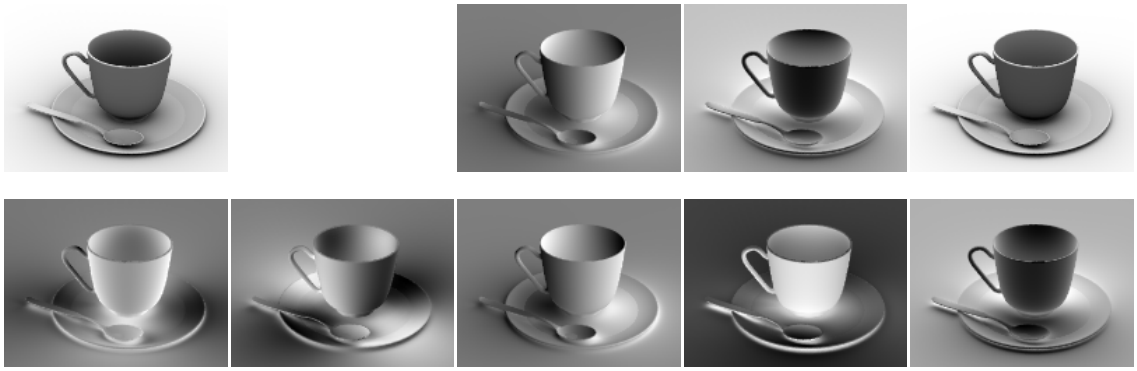


Figure 2.14: We show the first nine harmonic images created from more than three thousand directional source images derived from a 3D model of one teacup. The top row contains the zeroth harmonic (left) and the three first order harmonic images (right). The second row shows the images derived from the second harmonics. Image values are scaled to the full range of the intensity.

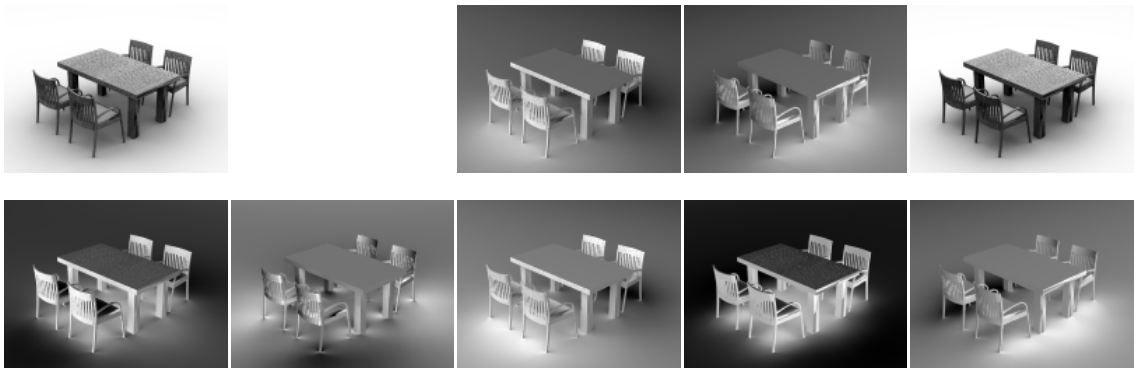


Figure 2.15: We show the first nine harmonic images created from more than three thousand directional source images derived from a 3D model of one table with four chairs. The top row contains the zeroth harmonic (left) and the three first order harmonic images (right). The second row shows the images derived from the second harmonics. Image values are scaled to the full range of the intensity.

tion from them. Figure 2.10 shows the sphere maps of four of the light probes used in our experiments. We considered a scene where the objects were placed on an infinite plane, so only lights coming from the upper hemisphere are taken into account.

Figure 2.16 provides the evaluation results of the teacup image in Figure 2.14. It shows ground truth images (row 1) and images obtained by using the method based on spherical harmonics (row 2), NNL with 100 directional sources (DS) (row 3), NNL with 300 DS (row 4), SDP (row 5), Haar wavelets (row 6), and our method (row 7). We recovered illumination distributions from input images (row 1) rendered with the sphere maps of kitchen, grace, campus, and building (Figure 2.10), respectively. The image approximated using spherical harmonics is obtained by projecting the image onto the harmonic subspace. It fails to capture the apparent shadows cast on the plate and ground. For NNL, we tested two versions, using the 100 and 300 largest DS respectively, from 977 possible ones. The reason is, as illustrated in Figure 2.16, NNL with 100 DS failed to generate a reasonable result. This tells us the results of NNL is not sparse and require a large number of directional sources in order to produce good results. Comparing with spherical harmonics, SDP captures more details of the cast shadows, but the shadows are very fuzzy and the shadow boundaries are unclear. We render the image with 102 Haar basis functions as in [103]. Both Haar wavelets and our method reproduce the shadows reliably.

Figure 2.17 provides the evaluation results of the teacup image in Figure 2.15. It achieves similar results to Figure 2.16. Our method recovers the illumination distribution reliably.

To quantitatively evaluate the performance of the methods in terms of speed and

Figure 2.16: Experiments on synthetic images rendered from one teacup: Ground truth images from different lighting probes as indicated (row 1), images recovered from different approaches spherical harmonics (row 2), NNL with 100 DS (row 3), NNL with 300 DS (row 4), SDP (row 5), Haar wavelets with 102 basis (row 6), and our method with 100 DS (row 6).

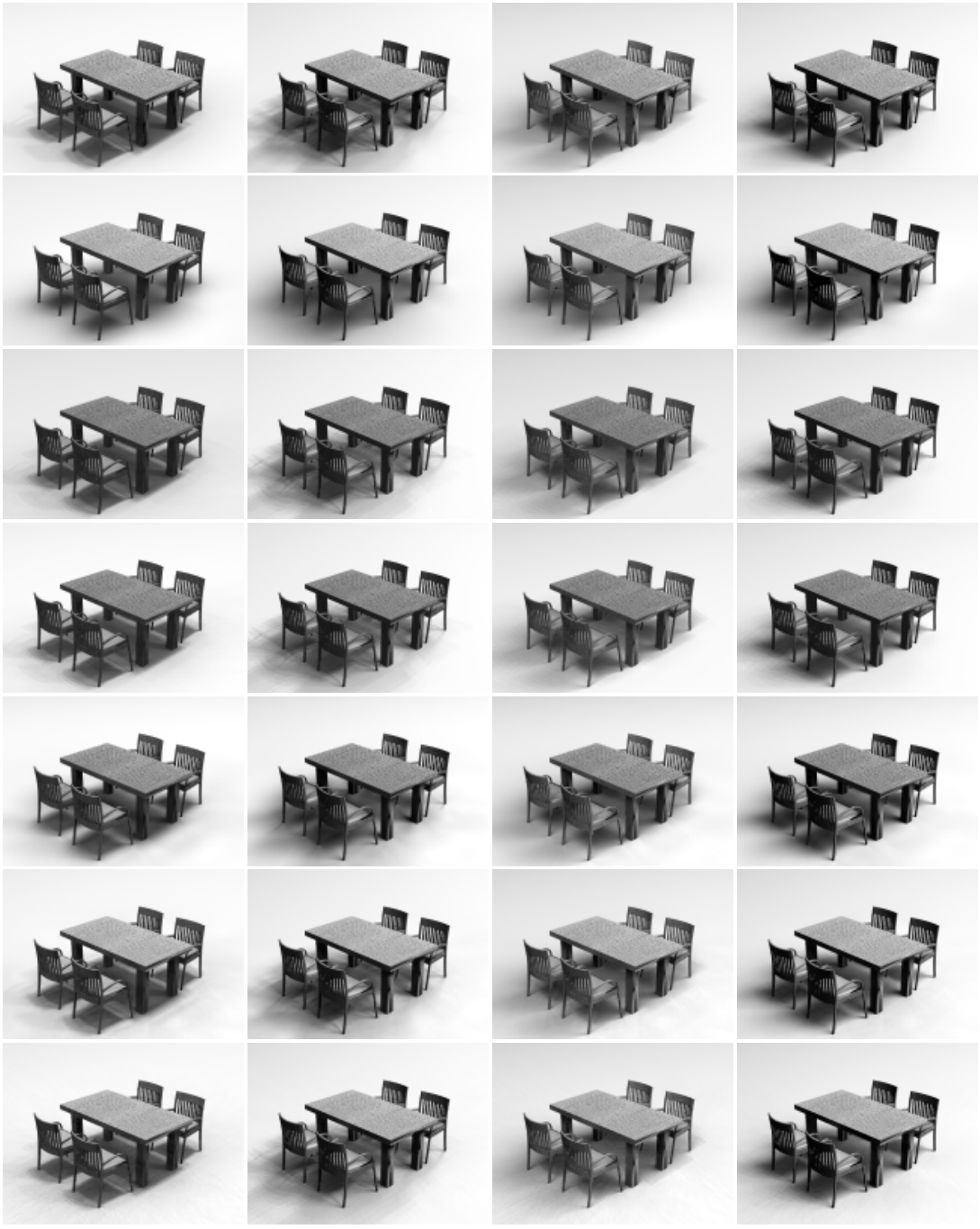


Figure 2.17: Experiments on synthetic images rendered from one table with four chairs: Ground truth images from different lighting probes as indicated (row 1), images recovered from different approaches spherical harmonics (row 2), NNL with 100 DS (row 3), NNL with 300 DS (row 4), SDP (row 5), Haar wavelets with 102 basis (row 6), and our method with 100 DS (row 6).

Table 2.1: RMS errors and average run time on the synthetic image dataset. Note: the running time does not include the preprocessing for generating "basis" images (same for Tables 2.3 and 2.4).

Method	Probe Kitchen	Probe Grace	Probe Building	Probe Campus	Avg. Run Time (sec.)
Spherical Harmonics	8.00	12.23	12.21	7.39	0.01
NNL (100 DS)	55.74	17.31	39.41	74.87	9389.8
NNL (300 DS)	5.96	2.80	1.87	12.50	9389.8
SDP	3.21	4.11	3.48	1.26	10.9
Haar Wav. (102 basis)	3.42	3.12	1.61	0.96	1322.0
Our method (100 DS)	2.33	2.69	1.22	1.09	11.8

accuracy, we measure the quality of the approximation by looking at RMS and the speed by run time. The errors in pixel values and run time in seconds are shown in Table 2.1, and Table 2.2. One can find that the error in our method is the smallest of all the listed methods and the run time is much smaller than the Haar wavelets method which has comparable accuracy to our method. Therefore, our method works best for recovering illumination from cast shadows in terms of both accuracy and speed.

Table 2.2: RMS errors and average run time on the synthetic image dataset. Note: the running time does not include the preprocessing for generating "basis" images (same for Tables 2.3 and 2.4).

Method	Probe Kitchen	Probe Grace	Probe Building	Probe Campus	Avg. Run Time (sec.)
Spherical Harmonics	11.09	9.44	9.36	8.64	0.01
NNL (100 DS)	15.26	10.91	10.84	12.20	9410.5
NNL (300 DS)	8.72	10.90	10.73	9.51	9410.5
SDP	7.29	8.36	7.62	6.49	11.2
Haar Wav. (102 basis)	5.92	4.04	4.43	5.20	1332.0
Our method (100 DS)	5.71	3.95	4.37	4.90	12.0

2.8.3 Experiments with Real Data

For real images, we conduct two kinds of experiments. First, all the algorithms are tested for illumination recovery tasks on *c*hair1 and *c*ouch. The results are shown in the left column of Figure 2.19 and in Figure 2.20. Second, we apply the recovered illumination from *c*hair1 to the model of *c*hair2 and then compare the results to the ground truth image. The second test is similar to those used for lighting recovery in [103]. The results are shown in the right column of Figure 2.19. The RMS errors and run time statistics are summarized in Tables 2.3 and 2.4.

Figure 2.18 show the first nine harmonic images created from more than three thou-

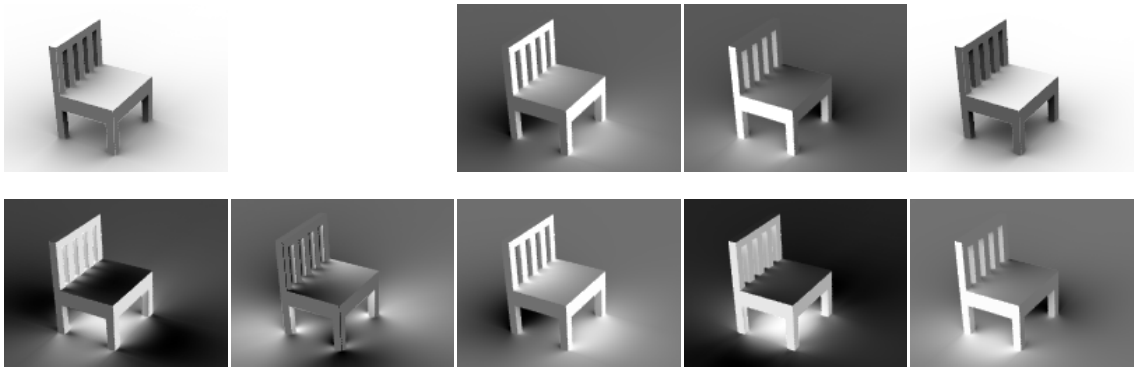


Figure 2.18: We show the first nine harmonic images created from more than three thousand directional source images derived from a 3D model of the chair1. The top row contains the zeroth harmonic (left) and the three first order harmonic images (right). The second row shows the images derived from the second harmonics. Image values are scaled to the full range of the intensity.

sand directional source images derived from a 3D model of chair1. The top row contains the zeroth harmonic (left) and the three first order harmonic images (right). The second row shows the images derived from the second harmonics. Image values are scaled to the full range of the intensity.

In Figure 2.19, the first row is the images of chair1 and chair2 which has the same lighting as chair1. The left column (chair1) and right column (chair2) show the images rendered with the lighting recovered from the chair1 image using spherical harmonics (row 2), NNL with 100 directional sources (row 3), SDP with 100 directional sources (row 4), Haar wavelets with 102 basis functions (row 5), and our method with 100 directional sources (row 6).

All these experiments show the superiority of our methods. Spherical harmonics fail to capture the apparent shadows cast on the seat of the chair and the ground. In

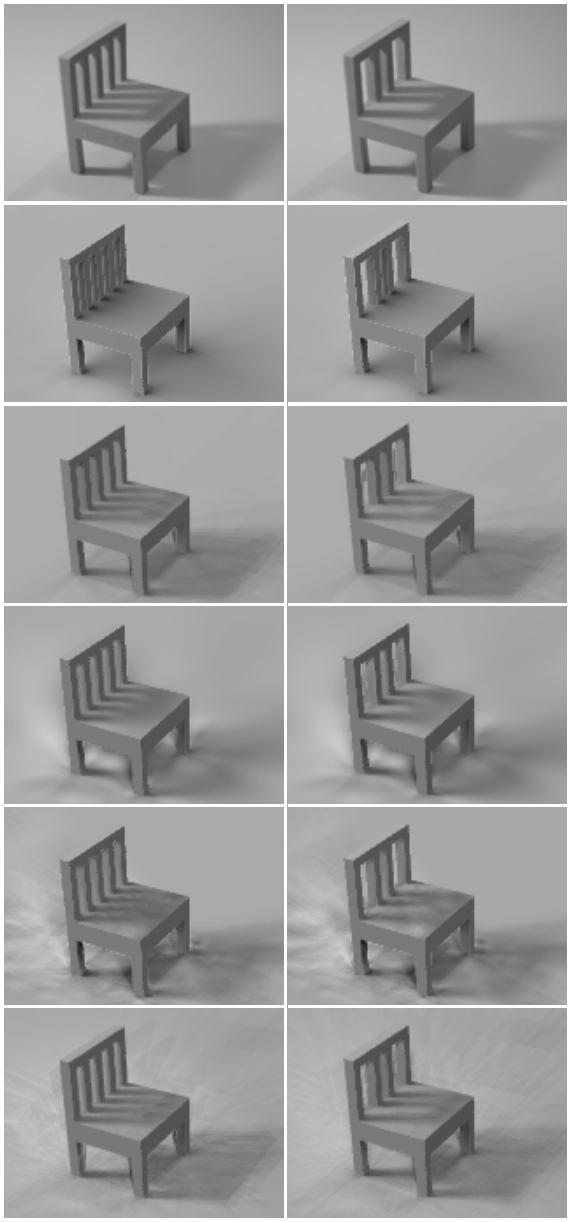


Figure 2.19: First row is the images of chair1 and chair2 which has the same lighting as chair1. The left column (chair1) and right column (chair2) show the images rendered with the lighting recovered from chair1 image using spherical harmonics (row 2), NNL with 100 directional sources (row 3), SDP with 100 directional sources (row 4), Haar wavelets with 102 basis functions (row 5), and our method with 100 directional sources (row 6).

comparison, SDP captures more details of the cast shadows, but the shadows are very fuzzy and there are some highlights on the ground. NNL can produce accurate shadows, but the shadows are intersecting and overlapping each other, causing the image to be unrealistic to the user. The Haar wavelets method produces accurate shadows, but there are some highlights on the ground. Our method generates visually realistic images and produces accurate shadows both on the seat and the ground. In addition, Table 2.3 shows the RMS error and run time for all the methods. Our method achieves the smallest error of all the methods in only tens of seconds run time. Figure 2.20 and Table 2.4 show the experimental results for the couch under natural indoor lighting. Again, our method achieves the best results in terms of speed and accuracy. Hence, it can be concluded that our method works reliably and accurately in recovering illumination and producing cast shadows for real images as well.

2.8.4 Sparsity Evaluation

In the previous section, we argue that we can approximate the query image well using a sparse set of directional light sources. To justify our argument, we conduct experiments on synthetic and real images. Figure 2.21 shows the RMS versus number of possible directional sources for synthetic images rendered with the grace light probe (left) and a real image (right) under natural indoor lighting. The accuracy improves gradually as the number of directional sources increases. From the plots, we can see after a certain number of directional sources (≈ 50 for the left and ≈ 180 for the right), the error remains constant. It matches the argument that we can approximate the query image well enough

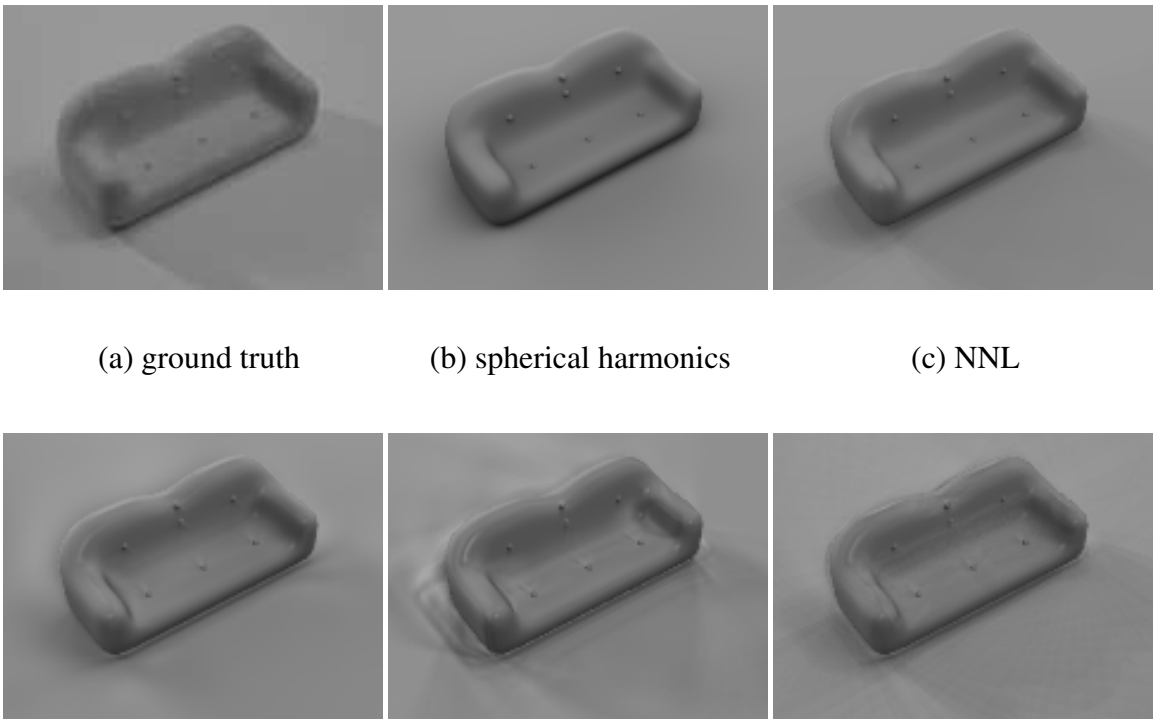
Table 2.3: RMS errors and run times on the real images of chair1 and chair2. The RMS for chair1 is for lighting recovery (Figure 2.19 left); while the RMS for chair2 is for lighting evaluation (Figure 2.19 right).

Method	Chair1 RMS Estimation	Chair2 RMS Evaluation	Run time (sec.)
Spherical Harmonics	13.99	15.31	0.01
NNL (100 DS)	10.26	10.35	1854.89
SDP	9.38	9.40	10.88
Haar Wav. (102 basis)	10.75	11.02	1529.60
Our method (100 DS)	7.50	8.24	14.54

Table 2.4: RMS errors and run times on real images for the couch.

Method	RMS	Run time (sec.)
Spherical Harmonics	9.39	0.01
NNL (100 DS)	7.37	2050.22
SDP	7.01	14.62
Haar Wav. (102 basis)	7.84	1585.27
Our method (100 DS)	6.56	13.82

using only a sparse set of directional sources and after a certain number of directional sources, increasing the number of directional sources does not improve the accuracy.



(a) ground truth (b) spherical harmonics (c) NNL

(d) SDP (e) Wavelets (f) our method

Figure 2.20: (a) Ground truth image of couch. (b)-(f) show the image rendered with the lighting recovered from (a) using different approaches, where (c) and (f) use 100 directional sources, and (e) uses 102 wavelet basis.

2.9 Conclusions

In this chapter, we start from describing basic shadow concepts and the spherical harmonic analysis on the Lambertian model we assume for the objects. Shadow concepts such as the umbra and the penumbra, attached and cast shadows are introduced and examples are given to illustrate the difference between them. Examples are also show the rich information provided by cast shadows. A spherical harmonic representation of Lambertian objects is briefly reviewed and its shortcomings are shown by a simple example using images with prominent cast shadows. We provide a simple example and explain

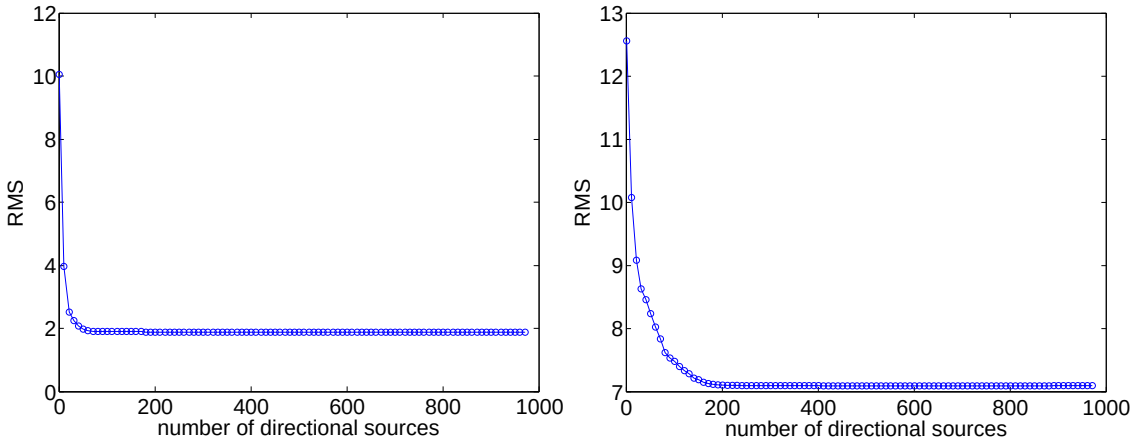


Figure 2.21: The improvement in accuracy by adding directional sources. RMS versus number of directional sources for a synthetic image rendered with grace light probe (left) and a real image in Figure 2.19 (first row chair2) (right) under natural indoor lighting.

that although the dimensionality of the subspace of images with cast shadows can go up to infinity, the illumination can still be well approximated by a sparse set of directional sources. Following this example, we derive a theoretical model and cast illumination recovery as an ℓ_1-regularized least squares problem. An efficient and fast solution is provided to find the most significant directional sources for the estimation. Experiments on both synthetic and real images have shown the effectiveness of our method in both accuracy and speed.

Chapter 3

Robust Visual Tracking using ℓ_1 Minimization

3.1 Introduction

In this chapter, we develop a robust visual tracking framework by casting the tracking problem as finding a sparse approximation in a template subspace. Motivated by the work in [137], we propose handling occlusion using trivial templates, such that each trivial template has only one non-zero element (see Figure 3.1). Then, during tracking, a target candidate is represented as a linear combination of the template set composed of both target templates (obtained from previous frames) and trivial templates. The number of target templates are far fewer than the number of trivial templates. Intuitively, a good target candidate can be efficiently represented by the target templates. This leads to a sparse coefficient vector, since coefficients corresponding to trivial templates (named trivial coefficients) tend to be zeros. In the case of occlusion (and/or other unpleasant issues such as noise corruption or background clutter), a limited number of trivial coefficients will be activated, but the whole coefficient vector remains sparse. A bad target candidate, on the contrary, often leads to a dense representation[1](e.g., Figure 3.4). The sparse representation is achieved through solving an ℓ_1-regularized least squares problem, which can be done efficiently through convex optimization. Then the candidate with the

[1]Candidates similar to trivial templates have sparse representations, but they are easily filtered out for their large dissimilarities to target templates.

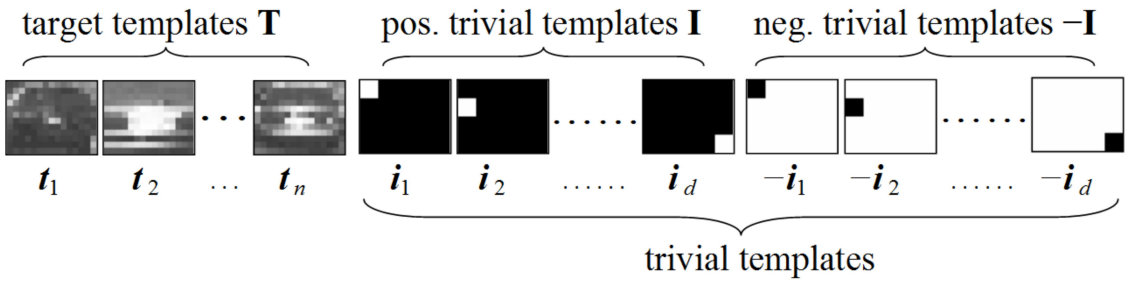


Figure 3.1: Templates used in our proposed approach (from the testing sequence, Figure 3.10).

smallest target template projection error is chosen as the tracking result. After that, tracking is led by the Bayesian state inference framework in which a particle filter is used for propagating sample distributions over time.

Three additional components are included in our approach to further improve robustness. First, the velocity estimation of horizontal and vertical translation parameters for the moving object is incorporated in the dynamic state transition model. This improves the performance by concentrating more samples on the true location of the target and use them more efficiently. Second, we enforce nonnegativity constraints to the sparse representation. These constraints are especially helpful to eliminate clutter that is similar to target templates with reversed intensity patterns. The constraints are implemented by including both positive and negative trivial templates in the template set. Third, we dynamically update the target template set to keep the representative templates throughout the tracking procedure. This is done by adjusting template weights by using the coefficients in the sparse representation. A flowchart of our proposed tracker is represented in Figure 3.2. We tested the proposed approach on numerous video sequences involving heavy occlusion, large illumination and pose changes. The proposed approach shows ex-

cellent performance in comparison with four previously proposed trackers, including the mean shift (MS) tracker [24], the covariance (CV) tracker [108], the appearance adaptive particle filter (AAPF) tracker [149], and the ensemble (ES) tracker [3].

Our work is motivated by recent advances in sparse representation [16, 33] and its application in computer vision. The most relevant work is [137] where sparse representation is applied for robust face recognition. Other applications include background subtraction [20], media recovery [47], texture segmentation [81], and lighting estimation [89], etc.

3.2 Motion Model

In the tracking framework, we apply an affine image warping to model the object motion of two consecutive frames. From (1.35) and (1.36), we have the inference formula:

$$p(x_t|y_{1:t}) \propto p(y_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{3.1}$$

where the y_t is the observation and x_t is the motion parameter of the target between the consecutive frames.

The tracking process is governed by the two important components: state transition model $p(x_t|x_{t-1})$ which models the temporal correlation of state transition between two consecutive frames, and observation model $p(y_t|x_t)$ which measures the similarity between the appearance observation and the approximation using the target model.

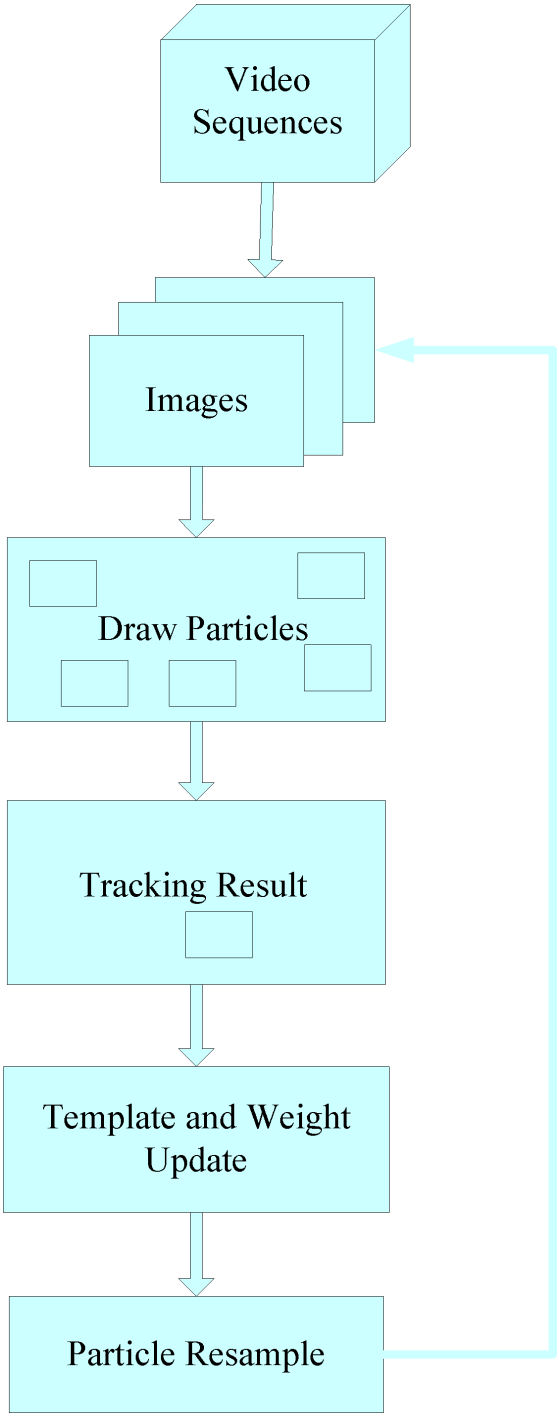


Figure 3.2: A flowchart of our proposed tracker.

3.2.1 State Transition Model

The state variable x_t is modeled by the six parameters of the affine transformation parameters $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, t_1, t_2)$ and velocity components (v_1, v_2), where $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$

are the deformation parameters, (t_1, t_2) are the 2D translation parameters, and (v_1, v_2) are the velocity of the horizontal and vertical translation parameters (t_1, t_2).

To propagate the particles, sometimes the velocity of the motion model is also taken into account [14, 15, 146]. The deformation parameters in x_t are modeled independently by a Gaussian distribution around the previous state x_{t-1}. We use an average velocity for the translation parameters to model the object motion.

$$\begin{aligned}(\alpha_1, \alpha_2, \alpha_3, \alpha_4, t_1, t_2)_t &= (\alpha_1, \alpha_2, \alpha_3, \alpha_4, t_1, t_2)_{t-1} + (0, 0, 0, 0, v_1, v_2)_{t-1} \cdot \Delta t + \epsilon_t \\ (v_1, v_2)_t &= \frac{\sum_{k=n}^{1} (v_1, v_2)_{t-k}}{n}\end{aligned} \tag{3.2}$$

The process noise ϵ_t for each state variable is independently drawn from zero-mean normal distributions. The variances of affine parameters are different and do not change over time. Δt is dependent on the frame-rate of the sequence. n is the number of previous velocities used for averaging. The larger the variances of the parameters, the more particles are needed to model the distribution. It is possible the target is tracked with better precision, but the cost of the computation to evaluate all the possible particles will be high as well. Some work [114, 149] exploits the balance between the efficient (less particles, less precision) and effective (more particles, better precision) in visual tracking.

3.2.2 Observation Model

By applying an affine transformation using x_t as parameters, we crop the region of interest y_t from the image and normalize it to be the same size as the target templates in the gallery. The observation model $p(y_t|x_t)$ reflects the similarity between a target candidate and the target templates. We assume y_t is generated from a subspace spanned

by the target template set. The probability of a sample generated from the subspace of template set, $p(y_t|x_t)$, is formulated from the error approximated by the target templates using ℓ_1 minimization. The likelihood of the projected sample is governed by a Gaussian distribution as follows:

$$p(y_t|x_t) = \mathbb{N}(y_t|\hat{y}_t, \Sigma) \tag{3.3}$$

where the mean of the Gaussian distribution is $\hat{y}_t$, which is the approximation by the target template set using ℓ_1 minimization. Σ is the covariance matrix. We assume the values are independent of each other and Σ turns out to be a diagonal matrix with only nonzero elements on the diagonal.

3.3 ℓ_1 Minimization Tracking

3.3.1 Sparse Representation of a Tracking Target

The global appearance of one object under different illumination and viewpoint conditions is known to lie approximately in a low dimensional subspace [78, 144]. Given target template set $\mathbf{T} = [\mathbf{t}_1 \dots \mathbf{t}_n] \in \mathbb{R}^{d \times n}$ ($d >> n$), containing n target templates such that each template $\mathbf{t}_i \in \mathbb{R}^d$ (we stack template image columns to form a 1D vector), a tracking result $\mathbf{y} \in \mathbb{R}^d$ approximately lies in the linear span of $\mathbf{T}$,

$$\mathbf{y} \approx \mathbf{T}\mathbf{a} = a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + \cdots + a_n\mathbf{t}_n \ , \tag{3.4}$$

where $\mathbf{a} = (a_1, a_2, \cdots, a_n)^\top \in \mathbb{R}^n$ is called a *target coefficient vector*.

In many visual tracking scenarios, target objects are often corrupted by noise or partially occluded. The occlusion creates unpredictable errors. It may affect any part of

the image and appear at any size on the image. To incorporate the effect of occlusion and noise, Equation 3.4 is rewritten as

$$\mathbf{y} = \mathbf{T}\mathbf{a} + \epsilon \,, \tag{3.5}$$

where ϵ is the error vector – a fraction of its entries are nonzero. The nonzero entries of ϵ indicate the pixels in $\mathbf{y}$ that are corrupted or occluded. The locations of corruption can differ for different tracking images and are unknown to the computer. Following the scheme in [137], we can use trivial templates $\mathbf{I} = [\mathbf{i}_1, \mathbf{i}_2, ..., \mathbf{i}_d] \in \mathbb{R}^{d \times d}$ to capture the occlusion as

$$\mathbf{y} = \begin{bmatrix} \mathbf{T}, & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \\ \mathbf{e} \end{bmatrix} \,, \tag{3.6}$$

where a *trivial template* $\mathbf{i}_i \in \mathbb{R}^d$ is a vector with only one nonzero entry (i.e. $\mathbf{I}$ is an identity matrix), and $\mathbf{e} = (e_1, e_2, \cdots, e_d)^\top \in \mathbb{R}^d$ is called a *trivial coefficient vector*.

3.3.2 Nonnegativity Constraints

In principle, the coefficients in $\mathbf{a}$ can be any real numbers if the target templates are taken without restrictions. However, we argue that in tracking, a tracking target can almost always represented by the target templates dominated by nonnegative coefficients. Here by "dominated" we mean that the templates that are most similar to the tracking target are positively related to the target. This is true when we start tracking from the second frame (the target is selected in the first frame manually or by a detection method), the target in the second frame will look more like the target in the first frame such that the coefficient is positive when the target in the first frame is used to approximate the

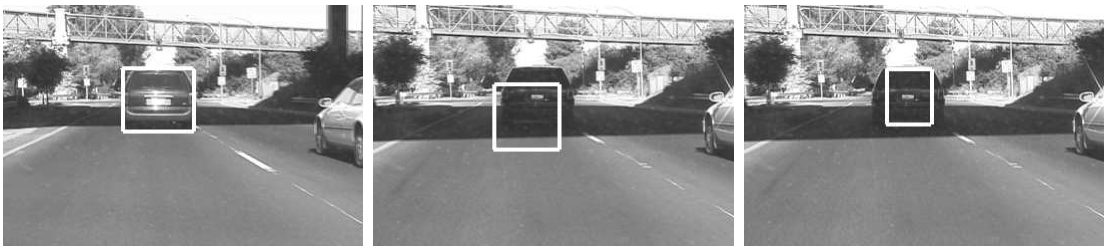


Figure 3.3: Left: target template. Middle: tracking result without non-negativity constraint. Right: tracking result with non-negativity constraint.

target in the second frame. In new frames the appearance of targets may change, but new templates will be brought in (may replace old templates) and the coefficients will still be positive for the most similar target templates in the following frames.

Another important argument for including nonnegative coefficients comes from their ability to filter out clutter that is similar to target templates at reversed intensity patterns, which often happens when shadows are involved. We give an example in Figure 3.3. In Figure 3.3, the left image shows the first frame in which the target template is created as in the bounding box. The middle image shows the tracking result without nonnegativity constraints, where the tracking result is way off the correct location. By checking the coefficients in **a**, we found that the failure is because the intensity pattern in the tracking result (dark in top and light in bottom) is roughly reversed compared to the target template (dark in bottom and light in top). This problem can be avoided by enforcing the nonnegativity constraints, as shown in the right image.

Enforcing nonnegativity constraints on the target coefficient vector **a** is straightforward. However, it is unreasonable to put such constraints directly on the trivial coefficient vector **e**. For this reason, we propose extending the trivial templates by including *negative*

trivial templates as well. Consequently, model (3.6) is now written as

$$\mathbf{y} = \begin{bmatrix} \mathbf{T}, & \mathbf{I}, & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{e}^+ \\ \mathbf{e}^- \end{bmatrix} \hat{=} \mathbf{Bc} \ , \quad \text{s.t. } \mathbf{c} \geq 0 \ , \tag{3.7}$$

where $\mathbf{e}^+ \in \mathbb{R}^d, \mathbf{e}^- \in \mathbb{R}^d$ are called a *positive* trivial coefficient vector and a *negative* trivial coefficient vector respectively, $\mathbf{B} = [\mathbf{T}, \mathbf{I}, -\mathbf{I}] \in \mathbb{R}^{d \times (n+2d)}$, and $\mathbf{c}^\top = [\mathbf{a}, \mathbf{e}^+, \mathbf{e}^-] \in \mathbb{R}^{n+2d}$ is a non-negative coefficient vector. Example templates are illustrated in Figure 3.1.

3.3.3 Achieving Sparseness through ℓ_1 Minimization

The system in (3.9) is underdetermined and does not have a unique solution for $\mathbf{c}$. The error caused by occlusion and noise typically corrupts a fraction of the image pixels. Therefore, for a good target candidate, there are only a limited number of nonzero coefficients in $\mathbf{e}^+$ and $\mathbf{e}^-$ that account for the noise and partial occlusion. Consequently, we want to have a sparse solution to (3.9). We exploit the compressibility in the transform domain by solving the problem as an ℓ_1-regularized least squares problem, which is known to typically yield sparse solutions [137]

$$\min ||\mathbf{Bc} - \mathbf{y}||_2^2 + \lambda ||\mathbf{c}||_1 \ , \tag{3.8}$$

where $||.||_1$ and $||.||_2$ denote the ℓ_1 and ℓ_2 norms respectively.

Figure 3.4 shows the coefficients approximated by the template set for the good and bad target candidates. The good and bad target candidates are shown in the red and blue bounding boxes on the top left image. The 10 target templates with size of 12×15 are

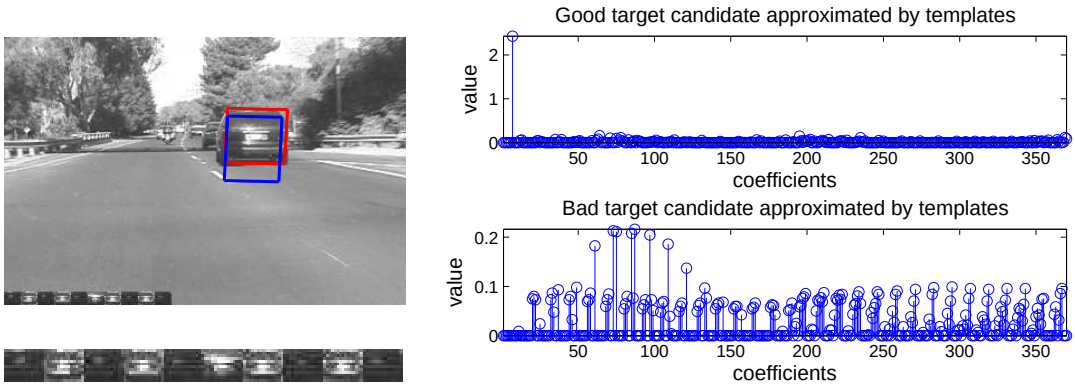


Figure 3.4: Top left: good and bad target candidates. Bottom left: Ten templates in the template set. They are the enlarged version of the templates shown on the bottom left corner of the top image. Top right: good target candidate approximated by template set. Bottom right: bad target candidate approximated by template set.

shown on the left corner, while the enlarged version is shown on the bottom left image. The images on the right show the good and bad target candidate approximated by the template set, respectively. The first 10 coefficients correspond to the 10 target templates used in the tracking and the remaining 360 coefficients correspond to the trivial templates. In the top right image, the seventh coefficient is relatively large compared to the remaining coefficients. Thus the seventh target template represents the good candidate well and the trivial templates are a small factor in approximating the good candidate. In the bottom right image, the coefficients are densely populated and trivial templates account for most of the approximation for a bad candidate in the left image.

Our implementation solves the ℓ_1-regularized least squares problem via an interior-point method based on [70]. The method uses the preconditioned conjugate gradients (PCG) algorithm to compute the search direction and the run time is determined by the product of the total number of PCG steps required over all iterations and the cost of a PCG step. We use the code from [27] for the minimization task.

We then find the tracking result by finding the smallest residual after projecting on the target template subspace, i.e., $||\mathbf{y} - \mathbf{Ta}||_2$. Therefore, the tracking result is the sample of states that obtain the largest probability, that is, the smallest error.

3.3.4 Template Update

Template tracking was suggested in the computer vision literature in [79], dating back to 1981. The object is tracked through the video by extracting a template from the first frame and finding the object of interest in successive frames. In [49], a fixed template is matched with the observations to minimize a cost function in the form of squared distance (SSD). A fixed appearance template is not sufficient to handle recent changes in the video, while at the other extreme, a rapidly changing model [125] which uses the best patch of interest in the previous frame, is susceptible to drift. Approaches have been proposed to overcome the drift problem [62, 67, 83] in different ways.

Intuitively, object appearance remains the same only for a certain period of time, but eventually the template is no longer an accurate model of the object appearance. If we do not update the template, the template cannot capture the appearance variations due to illumination or pose changes. If we update the template too often, small errors are introduced each time the template is updated. The errors are accumulated and the tracker drifts from the target. We tackle this problem by dynamically updating the target template set $\mathbf{T}$.

One important feature of ℓ_1 minimization is that it favors the template with larger norm because of the regularization part $||\mathbf{c}||_1$. The larger the norm of $\mathbf{t}_i$ is, the smaller

Algorithm 3 Template Update

1: **y** is the newly chosen tracking target.

2: **a** is the solution to (3.8).

3: **w** is current weights, such that $w_i \leftarrow ||\mathbf{t}_i||_2$.

4: τ is a predefined threshold.

5: Update weights according to the coefficients of the target templates. $w_i \leftarrow w_i * exp(a_i)$.

6: **if** ($sim(\mathbf{y}, \mathbf{t}_m) < \tau$), where sim is a similarity function. It can be the angle between two vectors or SSD between two vectors after normalization. $\mathbf{t}_m$ has the largest coefficient a_m, that is, $m = \arg\max_{1 \leq i \leq n} a_i$ **then**

7: $i_0 \leftarrow \arg\min_{1 \leq i \leq n} w_i$

8: $\mathbf{t}_{i_0} \leftarrow \mathbf{y}$, /*replace an old template*/.

9: $w_{i_0} \leftarrow$ median($\mathbf{w}$), /*replace an old weight*/.

10: **end if**

11: Normalize **w** such that $sum(\mathbf{w}) = 1$.

12: Adjust **w** such that $max(\mathbf{w}) = 0.3$ to prevent skewing.

13: Normalize $\mathbf{t}_i$ such that $||\mathbf{t}_i||_2 = w_i$.

coefficient a_i is needed in the approximation $||\mathbf{y} - \mathbf{Bc}||_2$. We exploit this characteristic by introducing a weight $w_i = ||\mathbf{t}_i||_2$ associated with each template $\mathbf{t}_i$. Intuitively, the larger the weight is, the more important the template is. At initialization, the first target template is manually selected from the first frame and zero-mean-unit-norm normalization is applied. The remaining target templates are created by perturbating one pixel in four possible directions at the corner points of the first template in the first frame. Thus we create

all the target templates (10 for our experiments) at the first frame. The target template set **T** is then updated with respect to the coefficients of the tracking result.

The updating in our approach includes three operations: template replacement, template updating, and weight updating. If the tracking result **y** is not similar to the current template set **T**, it will replace the least important template in **T** and be initialized to have the median weight of the current templates. The weight of each template increases when the appearance of the tracking result and template is close enough and decreases otherwise. The template update scheme is summarized in Algorithm 3.

3.4 Experiments

We implemented the proposed approach in MATLAB and evaluated the performance on numerous video sequences. The videos were recorded in indoor and outdoor environments at different format (color, grayscale, and IR) where the targets underwent lighting and scale changes, out-of-plane rotation, and occlusion. We choose different template sizes according to the width to height ratio for the target image in the first frame and, in all cases, the initial position of the target was selected manually. The other parameters for the ℓ_1 minimization template update and particle filter are the same from one experiment to the next.

3.4.1 Experimental Results

The first test sequence is obtained from http://www.cs.toronto.edu/~dross/ivt/. It shows a moving animal doll and presents challenging pose, lighting, and scale changes.

Figure 3.5 shows the tracking results using our proposed method, where the first row of each panel shows the tracked target which is enclosed with parallelogram. Five images on the second row from left to right are tracked target image patch, reconstructed and residue image using target templates, reconstructed and residue image using both target and trivial templates, respectively. The third and fourth rows show the ten target templates which are updated dynamically over time during the tracking. We can see more and more template images are replaced with the tracking results as tracking proceeds. The newly added templates are added since the old templates are no long able to capture the variation of the moving target and provide the updated appearance of the target so the tracker will not drift. The frame indices are 24, 48, 158, 222, 273 on the first row and 302, 383, 465, 515, 606 on the second row from left to right. In the first 200 frames (first four frames on the first row), the templates are not changed since the target undergoes out-of-plane rotation and translation, the appearance does not change much given the motion is modeled by affine transformation. After 200 frames, the target is moving farther and closer to the indoor light which causes the appearance of the target changes dramatically. Therefore, more and more tracking results are added to capture the appearance changes of the target. Our tracker tracks the target well throughout the whole probe sequence.

The second test sequence is obtained from PETS 2001 benchmark dataset http://www.cvg.cs.rdg.ac.uk/PETS2001/pets2001-dataset.html. Some samples of the tracking results are shown in Figure 3.6. The frame indices are 1442, 1479, 1495, 1498, 1505 on the first row, and 1535, 1628, 1640, 1668, 1714 on the second row from left to right. It shows a person walking from right bottom corner of the image to the left. The person is small in the image and undergoes deformable motion when he walks passing a pole

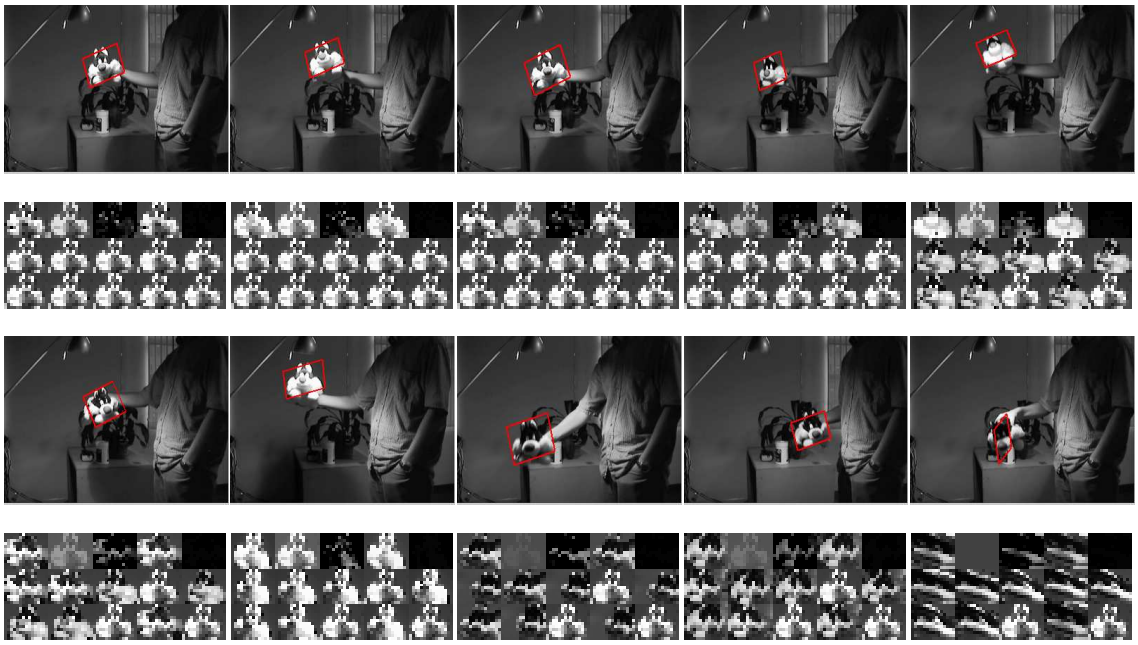


Figure 3.5: An animal doll moves with significant lighting, scale, and pose changes. The first row of each panel shows the tracked target which is enclosed with a parallelogram. The second row shows (from left to right) the tracked target image patch, reconstructed and residue images using target templates, reconstructed and residue images using both target and trivial templates. The third and fourth rows show the ten target templates.

(frame 1495, 1498, 1505) and long grasses (frame 1628, 1640, 1668). Since the person is thin in the image (the width of the target is small), the pole causes significant occlusion on the target. The template set gradually adds the tracking results as the person walks with hands swinging and legs spreading to maintain the variation of the template set, our tracker tracks well even through the significant motion and occlusion.

The third test sequence is a car moving very fast from close to far away. Some samples of the tracking results are shown in Figure 3.7. It shows significant scale changes and fast motion. When the car starts pulling away, it becomes smaller and smaller, and

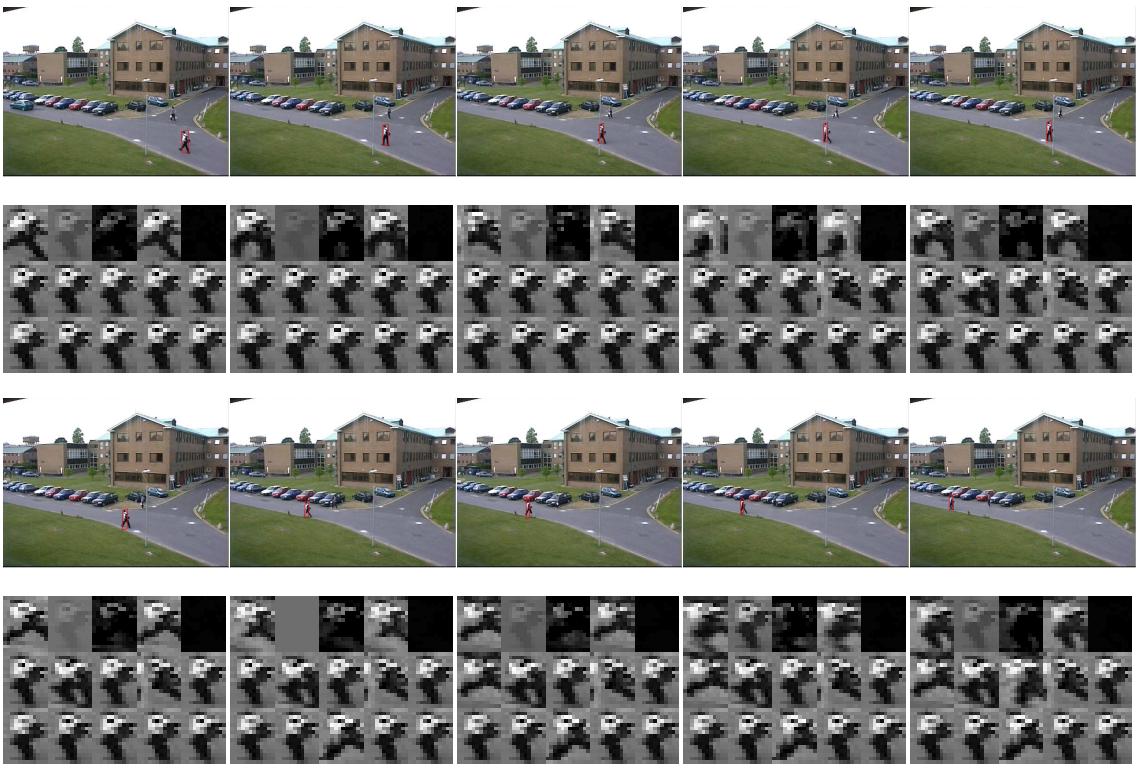


Figure 3.6: A person walks passing the pole and high grasses with significant body movements and occlusion. The first row of each panel shows the tracked target which is enclosed with a parallelogram. The second row shows (from left to right) the tracked target image patch, the reconstructed and residue images using target templates, reconstructed and residue images using both target and trivial templates. The third and fourth rows show the ten target templates.

harder to track since the target image is becoming smaller and gradually merges into the background. Even people have a hard time figuring out the precise location of the target. The frame indices are 547, 596, 634, 684, 723 on the first row, and 810, 859, 984, 1061, 1137 on the second row from left to right. Our tracker follows the car well throughout the sequence. The scale changes on the width and height go up to as much as 10 and 6 times.

The fourth test sequence shows a tank moving on the ground and is captured by

Figure 3.7: A car moves with significant scale changes and fast motion. The first row of each panel shows the tracked target which is enclosed with a parallelogram. The second row shows (from left to right) the tracked target image patch, reconstructed and residue images using target templates, reconstructed and residue images using both target and trivial templates. The third and fourth rows show the ten target templates.

a moving camera. Some samples of the tracking results are shown in Figure 3.8. The frame indices are 641, 671, 689, 693, 716 on the first row, and 737, 782, 820, 884, 927 on the second row from left to right. There is significant motion blur in the image and background clutter. The tank looks very similar to the environment around it, and the tracker very easily gets stuck to the ground and drifts away. The sudden movement of the camera poses great challenges to the tracker since the movement of the target is very unpredictable and the target can go any direction from the current location. The random sampled particles allow the tracker easily find the target's next movement. Therefore, our

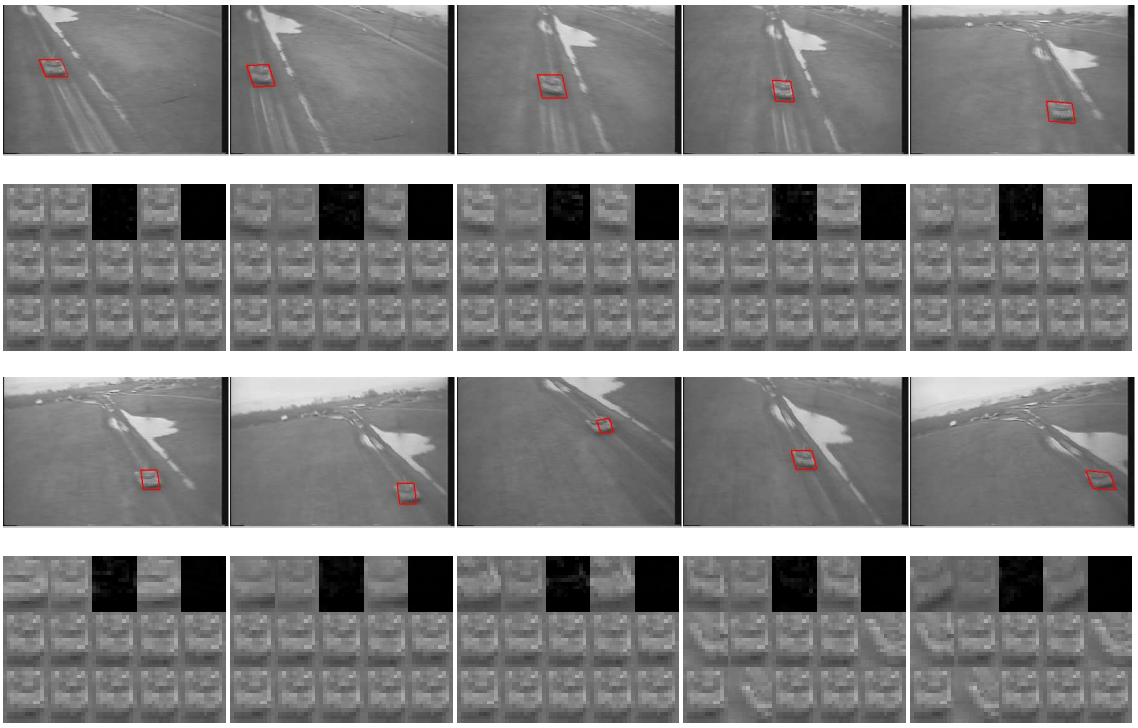


Figure 3.8: A moving tank, with video taken by a moving camera. There is significant motion blur in the image and background clutter. The first row of each panel shows the tracked targets, enclosed with a parallelogram. The second row shows (from left to right) tracked target image patch, reconstructed and residue images using target templates, reconstructed and residue images using both target and trivial templates. The third and fourth rows show the ten target templates.

tracker locks on the target very well throughout the whole probe sequence.

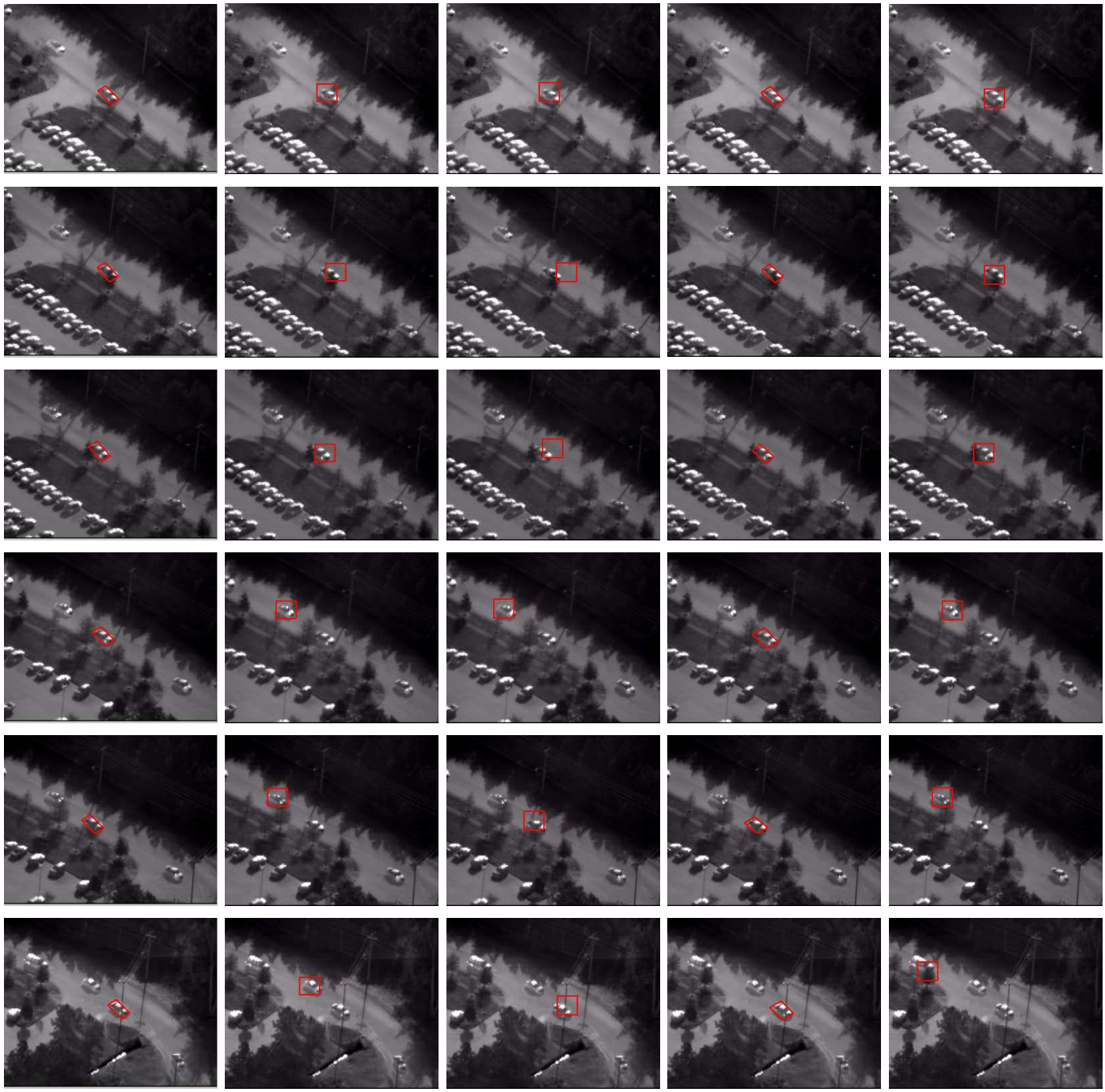
3.4.2 Qualitative Comparison

In our experiments, we compare the tracking results of our proposed method with those of a state-of-the-art standard Mean Shift (MS) tracker [24], covariance (CV) tracker [108], the appearance adaptive particle filter (AAPF) tracker [149], and the ensemble (ES)

tracker [3] on five of the sequences. The first two videos consist of 8-bit gray scale images while the last three are composed of 24-bit color images.

The first test sequence "pktest02" is an infrared (IR) image sequence from the VIVID benchmark dataset [23] PkTest02. Some samples of the final tracking results are demonstrated in Figure 3.9, where columns 1, 2, 3, 4, and 5 are for our proposed tracker, MS tracker, CV tracker, AAPF tracker, and ES tracker, respectively, in which six representative frames of the video sequence are shown. The frame indices are 1117, 1157, 1173, 1254, 1273, 1386. The target-to-background contrast is very low and the noise level is high for these IR frames. From Figure 3.9, we see that our tracker is capable of tracking the object all the time even with severe occlusions by the trees on the roadside. In comparison, MS and ES trackers lock onto the car behind the target starting from the fourth index frame. They keep tracking it in the rest of the sequences and are unable to recover it. CV tracker fails to track the target in the fourth index frame, similar to MS tracker, but it is able to recover the failure and track the target properly from the fifth index frame and throughout the rest of the sequence. In comparison, our proposed method avoids this problem and is effective under low contrast and in noisy situations. AAPF achieves similar results to our method.

The second test sequence "car4" is obtained from http://www.cs.toronto.edu/~dross/ivt/. The vehicle undergoes drastic illumination changes as it passes beneath a bridge and under trees. Some samples of the final tracking results are demonstrated in Figure 3.10, where columns 1, 2, 3, 4, and 5 are for our proposed tracker, MS tracker, CV tracker, AAPF tracker, and ES tracker, respectively. The frame indices are 181, 196, 233, 280, 308, 315. MS tracker loses the target very quickly and goes out of range from the fourth



(a) L1 (b) MS (c) CV (d) AAPF (e) ES

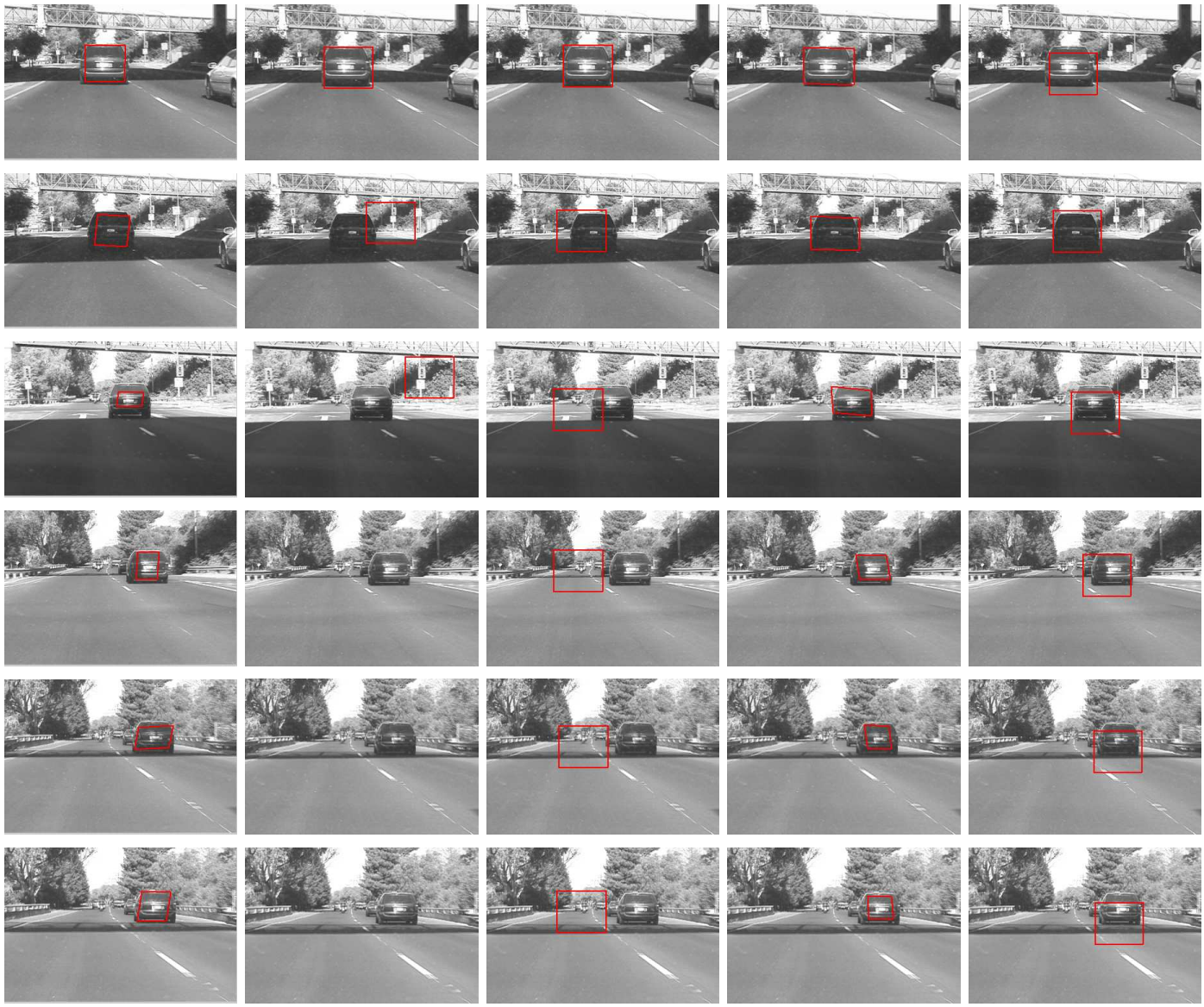
Figure 3.9: The tracking results of the first sequence: our proposed tracker (column 1), MS (column 2), CV (column 3), AAPF (column 4), and ES (column 5) over representative frames with severe occlusion.

index frame. CV tracker loses the target after tracking it for a while and gets stuck on the background. ES tracker tracks the car reasonably well given small scale changes in the sequence. Our tracker and AAPF are able to track the target well even though the drastic illumination changes.

The third test sequence "oneleaveshop" is obtained from http://groups.inf.ed.ac.uk /vision/CAVIAR/CAVIARDATA1/. In this video, the background color is similar to the color of the woman's trousers, and the man's shirt and pants have a similar color to the woman's coat. In addition, the woman undergoes partial occlusion. Some tracking result frames are given in Figure 3.11. The frame indices are 137, 183, 207, 225, 245, 271. It can be observed that MS, CV, and AAPF tracker start tracking the man when the woman is partially occluded at the third index frame, and are not able to recover the failure after that. ES tracker drifts away from the target very quickly and go out of the bounds of the image. Compared with other trackers, our tracker is more robust to the occlusion, which makes the target model not easily degraded by the outliers.

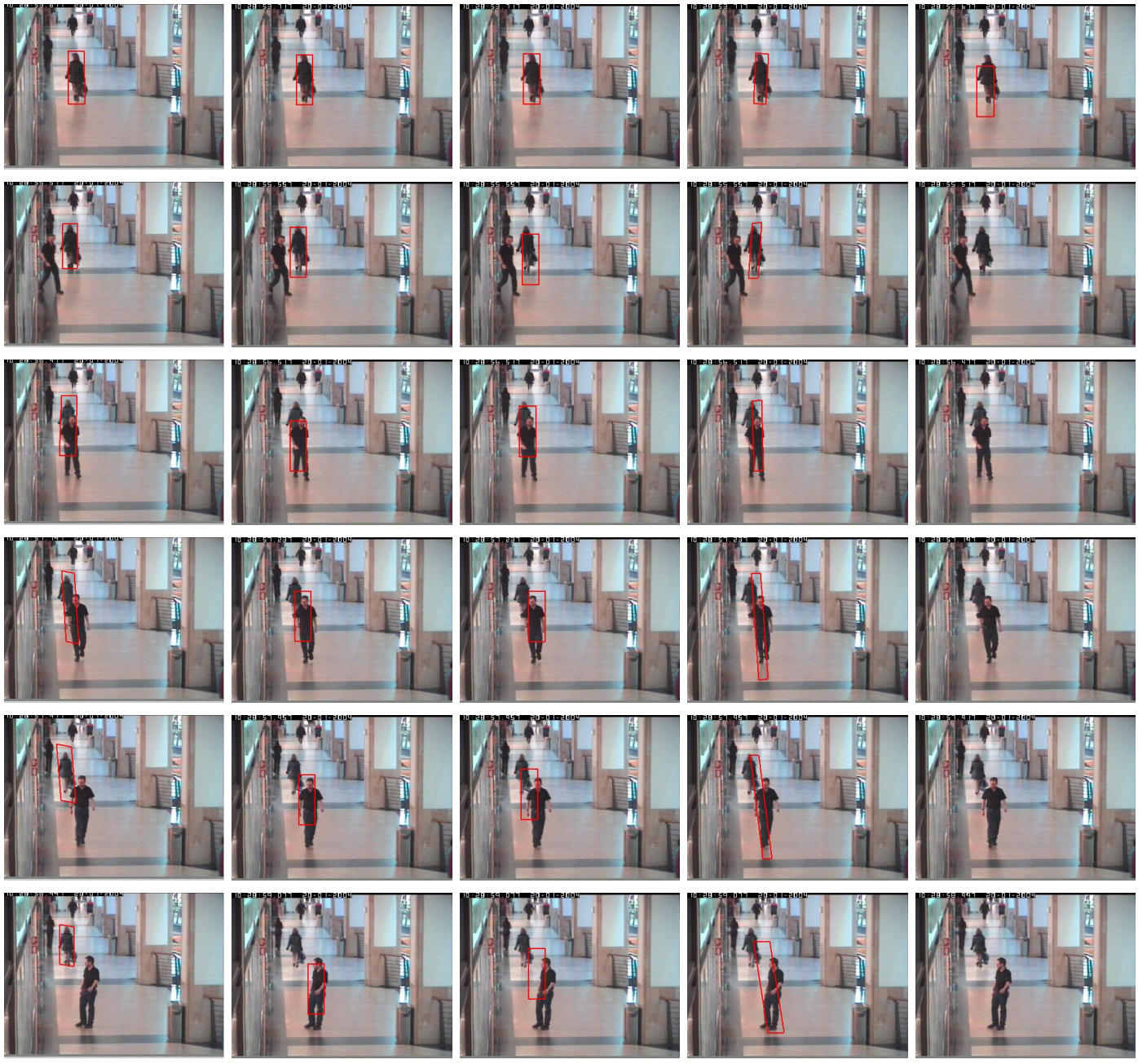
The fourth test sequence "birch_seq_mb" is obtained from http://vision.stanford.edu/ ~birch/headtracker/seq/. We show some samples of the tracking results for the trackers in Figure 3.12. The six representative frame indices are 422, 436, 448, 459, 466, and 474. The man's face is passing in front of the woman's face. Again, our method obtains good tracking results. The same for the MS and CV tracker. The AAPF tracker drifts apart when the severe occlusion happens in the second index frame. The ES tracker loses the target when the man's head occludes the girl's face and goes out of range of the image after that.

The fifth test sequence "V4V1_7_7" is an airborne car video. The car is running



(a) L1 (b) MS (c) CV (d) AAPF (e) ES

Figure 3.10: The tracking results of the second sequence: our proposed tracker (column 1), MS (column 2), CV (column 3), AAPF (column 4), and ES (column 5) over representative frames with drastic illumination changes.



(a) L1 (b) MS (c) CV (d) AAPF (e) ES

Figure 3.11: The tracking results of the third sequence: our proposed tracker (column 1), MS (column 2), CV (column 3), AAPF (column 4), and ES (column 5) over representative frames with partial occlusion and background clutter.

(a) L1 (b) MS (c) CV (d) AAPF (e) ES

Figure 3.12: The tracking results of the fourth sequence: our proposed tracker (column 1), MS (column 2), CV (column 3), AAPF (column 4), and ES (column 5) over representative frames with severe occlusion.

on a curved road and passing beneath the trees. It undergoes heavy occlusions and large pose changes. We show some samples of the tracking results for the trackers in Figure 3.13. The six representative frames indices are 215, 331, 348, 375, 393, and 421. MS tracker loses the target very quickly and goes out of range in the sixth frame. ES tracker drifts away from the target when it goes under the trees along the road and is heavily occluded. Although CV and AAPF can track the target, they do not locate the target well. Our tracker tracks the target very well throughout the whole sequence.

3.4.3 Quantitative comparison

To quantitatively compare robustness under challenging conditions, we show how many frames these methods can track the targets before the tracker fails. That is, after this frame, a tracker cannot recover without re-initialization. Table 3.1 shows the comparison between different methods on the 5 video sequences shown in the previous section. The number of total frames and the number of frames where the occlusion happens in each sequence are also shown in Table 3.1. The comparison demonstrates that our tracker performs more robustly than other trackers.

We also manually labeled the ground truth of the sequences "pktest02", "car4", "oneleavesshop", "birch_seq_mb", and "V4V1_7_7" from the previous section for 300, 300, 150, 93, and 300 frames, respectively. The evaluation criteria of the tracking error are based on the relative position errors (in pixel) between the center of the tracking result and that of the ground truth. Ideally, the position differences should be around 0.

As shown in Figure 3.14, the position differences of the results in our ℓ_1 tracker are

(a) L1 (b) MS (c) CV (d) AAPF (e) ES

Figure 3.13: The tracking results of the fifth sequence: our proposed tracker (column 1), MS (column 2), CV (column 3), AAPF (column 4), and ES (column 5) over representative frames with heavy occlusion and large pose variation.

Table 3.1: Comparison of different methods in terms of tracked frames.

	Frames	Occlusions	MS	CV	AAPF	ES	Ours
pktest02	300	216	51	243	300	118	300
car4	300	0	43	85	300	300	300
oneleavesshop	150	50	62	80	77	37	150
birch_seq_mb	93	41	93	93	62	37	93
V4V1_7_7	300	94	171	300	295	181	300

much smaller than those of the other trackers. It demonstrates the advantage of our ℓ_1 tracker.

3.5 Simultaneous Tracking and Recognition

In this section, we extend our ℓ_1 tracker and propose a simultaneous tracking and recognition method using ℓ_1 minimization in a probabilistic framework. When we locate the moving target in the image, we want to identify it based on the template images in the gallery. Typically, tracking is solved before recognition. When the object is located in the frame, it is cropped from the frame and transformed using an appropriate transformation. This tracking and recognition are performed sequentially and separately to resolve the uncertainty in the video sequences. The recognition after tracking strategy poses some difficulties such as selecting good frames and estimation of parameters for registration. We propose simultaneous tracking and recognition and apply it for the vehicle classifica-

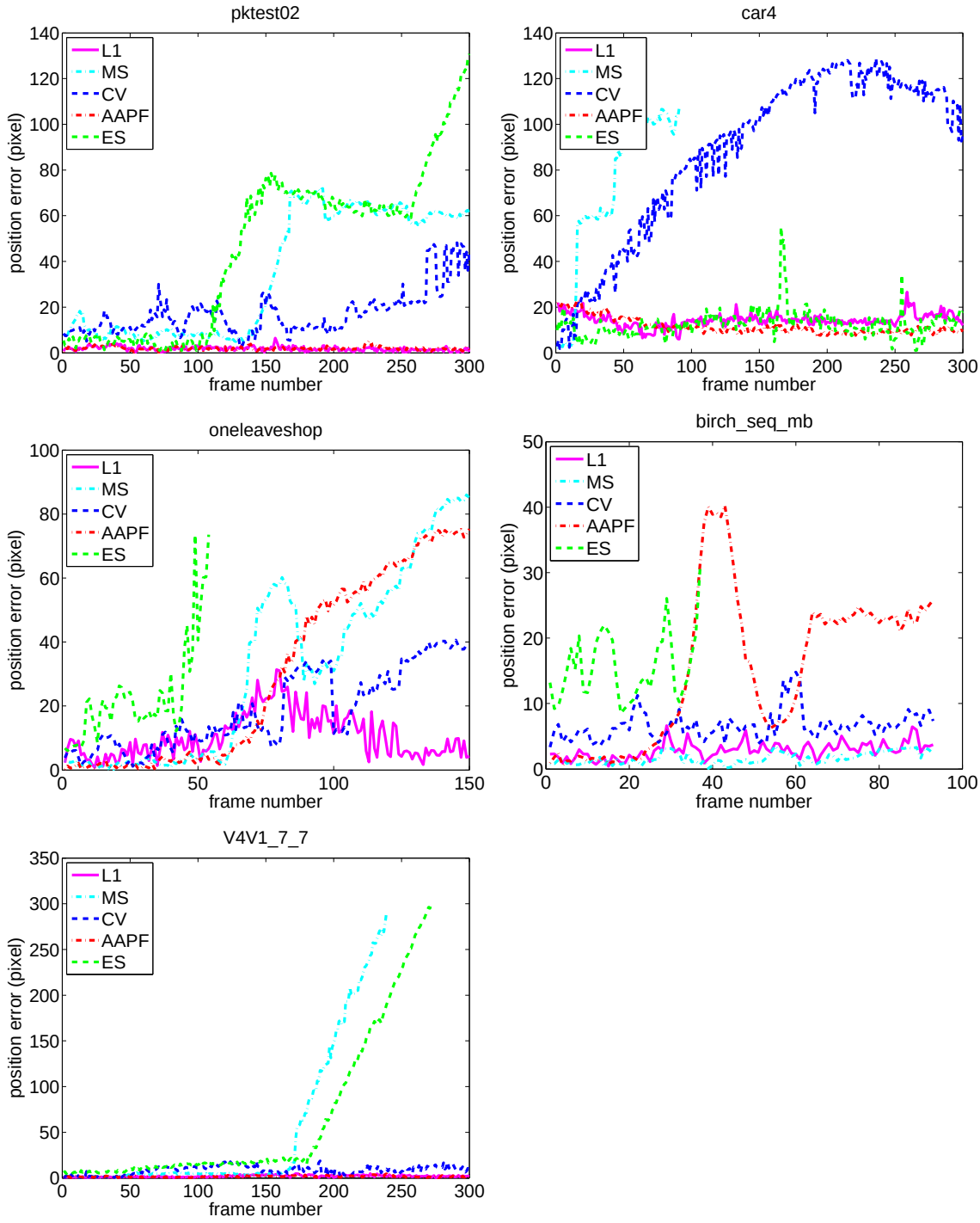


Figure 3.14: Quantitative comparison of the trackers in terms of position errors (in pixel).

tion in IR-based video sequences.

3.5.1 Algorithm Overview

In the previous section, we show the tracking result y can be written as

$$\mathbf{y} = \begin{bmatrix} \mathbf{T}^{(d)}, & \mathbf{I}, & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{e}^{+} \\ \mathbf{e}^{-} \end{bmatrix} \hat{=} \mathbf{Bc} \ , \quad \text{s.t. } \mathbf{c} \geq 0 \ , \tag{3.9}$$

where $\mathbf{T}^{(d)}$ is the target template set. We name it $\mathbf{T}^{(d)}$ because the template set is updated dynamically according to the template updating scheme. $\mathbf{e}^{+} \in \mathbb{R}^{d}, \mathbf{e}^{-} \in \mathbb{R}^{d}$ are called a *positive* trivial coefficient vector and a *negative* trivial coefficient vector, respectively, $\mathbf{B} = [\mathbf{T}^{(d)}, \mathbf{I}, -\mathbf{I}] \in \mathbb{R}^{d \times (n+2d)}$, and $\mathbf{c}^{\top} = [\mathbf{a}, \mathbf{e}^{+}, \mathbf{e}^{-}] \in \mathbb{R}^{n+2d}$ is a non-negative coefficient vector.

To extend the work to simultaneous tracking and recognition, we add additional components to the template set and name them as *static* templates $\mathbf{T}^{(s)}$. The templates in the $\mathbf{T}^{(s)}$ are not changing over time during the tracking and remain static throughout the process. The static template set $\mathbf{T}^{(s)}$ has two purposes: improve the tracking and provide recognition. It is defined as $\mathbf{T}^{(s)} = [\mathbf{T}_1^{(s)}, \mathbf{T}_2^{(s)}, \cdots, \mathbf{T}_m^{(s)}]$, where m is the number of object classes and $\mathbf{T}_i^{(s)}$ is the template set of the i^{th} object class.

We introduce an identity variable o_t which runs from object 1 to object m. The problem of simultaneous tracking and recognition is to infer the motion state variable x and identity variable o_t given the observations $y_{1:t}$ from frame 1 to t. We rewrite (1.36) as follows

$$p(x_t, o_t|y_{1:t}) = \frac{p(y_t|x_t, o_t)p(x_t, o_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{3.10}$$

$p(x_t, o_t|y_{1:t-1})$ can be recursively computed as follows:

$$p(x_t, o_t|y_{1:t-1}) = \int p(x_t, o_t|x_{t-1}, o_{t-1})p(x_{t-1}, o_{t-1}|y_{1:t-1})dx_{t-1} \tag{3.11}$$

We assume the x_t and o_t are independent of each other and $p(o_t|o_{t-1}$ is a uniform distribution. The above equation can be further rewritten as

$$\begin{aligned} p(x_t, o_t|y_{1:t-1}) &= \int p(x_t|x_{t-1})p(o_t|o_{t-1})p(x_{t-1}, o_{t-1}|y_{1:t-1})dx_{t-1} \\ &\propto \int p(x_t|x_{t-1})p(x_{t-1}, o_{t-1}|y_{1:t-1})dx_{t-1} \end{aligned} \tag{3.12}$$

For each class i, let $\delta_i : \mathbb{R}^{n_d+n_s+2d} \rightarrow \mathbb{R}^{n_d+n_s+2d}$ be the characteristic function that selects the coefficients associated with the dynamic template set and the ith class templates in the static template set. For x, $\delta_i(x)$ is a new vector whose only nonzero entries are the entries in x that are associated with the dynamic template set and the ith class in the static templates set. Using only the coefficients associated with the dynamic template set and the ith class, one can approximate the given tracking candidate y as $\hat{y}_i = B\delta_i(x)$. We then obtain the probability of the tracking result based on the object class identity and the motion state variable:

$$p(y|x, o) = p(y - B\delta_i(x)) \tag{3.13}$$

$p(y - B\delta_i(x))$ is a probability function defined by the tracking candidate y and the approximated result projected onto the dynamic template set and ith class static template set.

The sample has the maximum $p(y_t - B\delta_i(x_t^j))$ that implicitly encodes the tracking result (jth sample position) and recognition result (ith object class). The number of the dynamic templates in $\mathbf{T}^{(d)}$ can be the same as we proposed in [90] or fewer given the introduction of the static template set $\mathbf{T}^{(s)}$. The static template set $\mathbf{T}^{(s)}$ consists of the templates from m classes which are obtained from the training video sequences. The weights $\mathbf{w}$ are only assigned to the dynamic template set $\mathbf{T}^{(d)}$ which are dynamically updated over time during the tracking. No weights are assigned to the static template set $\mathbf{T}^{(s)}$ since the static templates are used for recognition and not updated over time. The template updating scheme is defined the same in the template update section. The norm of the dynamic template set is also updated over time according to the template update scheme, while the norm of the static template set is initialized to be $1/n_d$, where n_d is the number of templates in the dynamic template set, and kept constant during the whole tracking process. In the tracking process, we also introduce the tracking evaluation which would evaluate the performance of the tracking and prevent further tracking when the tracker drifts away from the target. The tracking evaluator gives a confidence score at each frame, and if the confidence value is below some threshold, restarts the whole simultaneous tracking and recognition process. The simultaneous tracking and recognition algorithm is summarized in Algorithm 4.

3.5.2 Tracking Evaluation

Most practical tracking systems often fail under some situations. This could be either because of illumination changes, pose variation or occlusion. Therefore, the need

Algorithm 4 Simultaneous tracking and recognition

1: **Input**: a matrix of dynamic, static and trivial templates, $\mathbf{B} = [\mathbf{T}^{(d)}, \mathbf{T}^{(s)}, \mathbf{I}, -\mathbf{I}] \in \mathbb{R}^{n_d+n_s+2d}$.

2: Initialize the weight $\mathbf{w}$ to be uniform for the dynamic templates. Each is set to be $1/n_d$. The recognition probability for each class is set to be uniform $p_i(0) = 1/m$.

3: The norm of the templates in $\mathbf{T}^{(d)}$ and $\mathbf{T}^{(s)}$ is set to be $1/n_d$.

4: **for** $t = 1$ to number of frames **do**

5: Solve the ℓ_1 minimization problem for each drawing samples $\min ||\mathbf{Bc} - \mathbf{y}_t||_2^2 + \lambda||\mathbf{c}||_1$, where $c \geq 0$.

6: Calculate the probability for $p(\mathbf{y}_t - \mathbf{B}\delta_i(\mathbf{x}_t^j))$ and pick the one with the largest probability. The ith class is the recognition result and the jth sample is the tracking result for the current frame.

7: Pick j as the tracking result and normalize $p(y_t - \mathbf{B}\delta_i(\mathbf{x}_t^j))$ across the i such that $\sum_{i=1}^{m} p(\mathbf{y}_t - B\delta_i(\mathbf{x}_t^j)) = 1$.

8: Multiply the recognition result with the one from previous frame and gives the recognition result for the frame from 1 to t. $p_i(t) = p_i(t-1) \times \max\, p(\mathbf{y}_t - \mathbf{B}\delta_i(\mathbf{x}_t^j))$.

9: Template updating.

10: Tracking evaluation.

11: **if** tracking confidence ¡ η, where η is a predefined threshold. **then**

12: restart tracking and recognition process.

13: **end if**

14: **end for**

for automatic performance evaluation emerges in these applications. Figure 3.15 shows the tracking result after running the tracker for some time. The bounding box is so large that one concludes that the tracker is already failing. Hence, evaluation is necessary to help us terminate tracking and restart the tracking and classification sequence. In the following section, we briefly review the tracking evaluation algorithm we proposed in [87, 88].

Figure 3.15: The vehicle is off tracking.

Our evaluation algorithm is based on measuring the appearance similarity and tracking uncertainty. The following features are examined in our evaluation:

1. Trace complexity q_{tc}: We define the trace complexity as the ratio of the curve length and straight length between the target centroids in different frames.

2. Motion step q_{ms}: It is defined as the distance between the box centers in two consecutive frames.

3. Scale change q_{sc}: To examine changes in object scale, we use two clues. One is the ratio of the current area to the initial area, the other is the scale change velocity.

4. Shape similarity q_{ss}: The change in the aspect ratio of the bounding box is also useful in providing some information about the object shape. It is defined as the ratio of the current aspect ratio over the initial ratio.

5. Appearance change q_{ac}: Three measures are used in our algorithm, the first one is the absolute pixel by pixel change between the current frame and the initial frame, the second one is the histogram difference between the current frame and the initial frame and the last one is related to the tracking algorithm over which the proposed algorithm was tested.

To obtain a comprehensive measure of the tracking performance, we combine the above five indicators. We first use empirical thresholds to find whether the tracker is uncertain according to the above five metrics, then we sum the five indicators using different weights to arrive at a confidence measure q. If the sum drops below some threshold, we conclude that the tracking performance is poor and needs re-initialization.

$$q = \sum_{j \in J} w_j I[q_j < \lambda_j],\ J \in \{tc, ms, sc, ss, ac\} \tag{3.14}$$

where w_j and λ_j are the corresponding weights and thresholds for the evaluation.

3.5.3 Experimental Results

In this section, we apply the simultaneous tracking and recognition on the IR-based vehicle classification. There are total of 4 different vehicles in the experiment. Figure 3.16 shows the static vehicle template set with five templates for each vehicle. They are called “bmp”, “m60”, “brdm”, and “wetting”, respectively. Figure 3.17 shows the

tracking and classification results of the vehicle "wetting". Figure 3.17 (a)-(f) show the tracking results for the index frames 440, 504, 701, 780, 1147, 1338, respectively. Figure 3.17 (g) shows the recognition scores for each vehicle and (h) shows tracker confidence q which is evaluated at each frame. The recognition score for "wetting" starts at 0.25 which is uniform between all the available vehicles and is gradually increasing to around 0.9 as the frame ends. This gives a high confidence to classify the moving vehicle in the video to be "wetting" which is correct. In Figure 3.17 (h), we set the confidence threshold q to 0.4 and the tracking and classification restarts around frame 750 when the tracking confidence level goes under 0.4. We did the experiment on four videos each with different vehicle and all the vehicles are classified correctly using our proposed method.

Figure 3.16: Static templates for four vehicles.

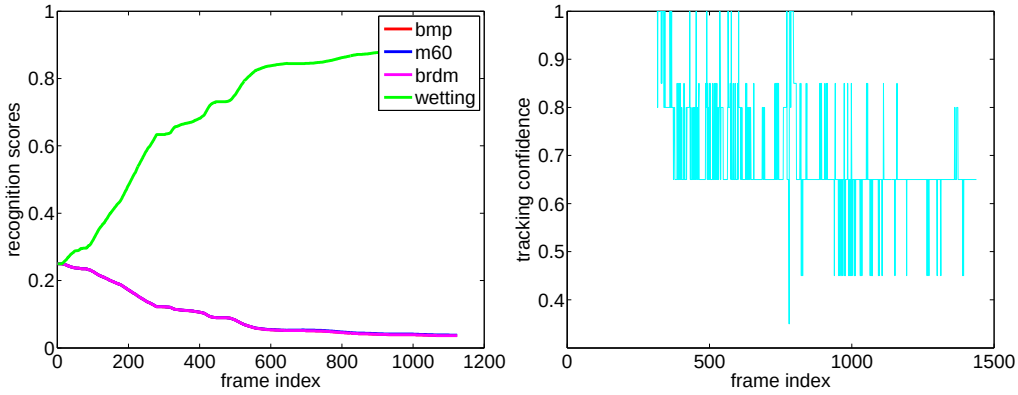
3.6 Conclusion

In this chapter we propose using a sparse representation for robust visual tracking. We model tracking as a sparse approximation problem and solve it through an ℓ_1-regularized least squares approach. For further robustness, we introduce nonnegativity constraints and dynamic template updating in our approach. In thorough experiments involving numerous challenging sequences and four other state-of-the-art trackers, our approach demonstrates very promising performance. We also extend our work to simultaneous tracking and recognition and apply it to the IR-based vehicle classification. A

tracking evaluation is introduced and conducted at each frame to give a tracking confidence, and the tracking and recognition process is restarted if the confidence level is below a certain predefined threshold. The experimental results demonstrate the effectiveness of our proposed method.

(a) (b) (c)

(d) (e) (f)



(g) (h)

Figure 3.17: Tracking and classification results of the vehicle “wetting”. (a)-(f) show the tracking results for the index frames 440, 504, 701, 780, 1147, 1338, respectively. (g) shows the recognition scores for each vehicle. (h) shows tracker confidence q which is evaluated at each frame.

Chapter 4

Summary and Future Research Directions

The dissertation addresses two fundamental problems in computer vision area: illumination estimation and visual tracking. The framework we propose essentially cast the illumination estimation and visual tracking as a sparse representation problem so that the problem can be solved through an ℓ_1-regularized least squares approach.

The illumination estimation method proposes to represent the lighting by two parts: 9D spherical harmonics (low frequency signal) and directional light sources (high frequency signal). Our focus is finding the fewest directional light sources to best approximate the illumination. Using the ℓ_1 minimization approach, the number of directional light sources to approximate the shadows is greatly reduced and the time to search the best representative sources is dramatically improved.

The visual tracking method proposes to approximate the target image by the target templates and trivial templates. The trivial templates account for occlusion, background clutter, and other unexpected noise. The problem fits to the sparse representation model given the target templates are greatly outnumbered by the trivial templates. We can infer the quality of the target candidate by examining the coefficients of the target templates since the coefficients of the trivial templates tend to be zeros while the target templates account most for the approximation. By enforcing sparse approximation, we tend to find out the most representative target templates and resist to the defectors such as occlusion

and noise.

The dissertation can be summarized as follows:

- An efficient modeling of the illumination estimation and visual tracking is proposed under the sparse representation framework.
- Each method is compared with several other state-of-the-art methods to show the robustness and efficiency of our proposed method.
- For visual tracking, further improvements are proposed including nonnegativity constraints for template coefficients and a dynamic template updating scheme.
- A simultaneous tracking and recognition method is proposed by introducing a set of static templates in addition to the templates used in visual tracking.

Our work can be extended in a number of ways. Here we briefly summary them:

- In-depth study of the ℓ_1 minimization will lead to better understanding of its performance and limitations. Efficient modeling and sparse representation can also benefit other computer vision areas.
- The visual tracking approach can be exploited to solve full occlusion and recapturing problems. Occlusion is detected by examining the value and distribution of the coefficients of the trivial templates.
- The background information can be incorporated in the tracking and further improve the robustness of the method.

- Speed up the computation in ℓ_1 minimization for visual tracking given the special structure of measurement matrix A, which has one positive and negative identity matrix.

- Further improvement on the speed of the illumination estimation method can be applied in lighting design, an interesting topic in computer graphics.

- Enhance the ℓ_1 tracker to work on the video sequence with out-of-plane rotation.

In a nutshell, the illumination estimation and visual tracking can be further studied using the framework and methods proposed in this dissertation.

Bibliography

[1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. "Structured importance sampling of environment maps", *SIGGRAPH*, 22(3):605-612, 2003.

[2] B. Anderson, and J. Moore. "Optimal Filtering", *New Jersey: Prentice Hall, Englewood Cliffs*, 1979.

[3] S. Avidan. "Ensemble Tracking", *CVPR*, 494-501, 2005.

[4] R. Azuma. "A survey of augmented reality. In Com- puter Graphics", *SIGGRAPH 95 Proceedings, Course Notes 9: Developing Advanced Virtual Reality Applications*, 1995.

[5] B. Babenko, M. Yang, and S. Belongie. "Visual Tracking with Online Multiple Instance Learning", *CVPR*, 2009.

[6] S. Baker and I. Matthews. "Lucas-kanade 20 years on: A unifying framework", *IJCV*, 56:221-255, 2004.

[7] R. Basri and D. Jacobs. "Lambertian Reflectances and Linear Subspaces", *PAMI*, 25(2):218-233, 2003.

[8] P. Belhumeur and D. Kriegman. "What is the Set of Images of an Object Under All Possible Lighting Conditions?", *IJCV*, 28(3):245-260, 1998.

[9] M.J. Black and D.J. Fleet, "Probabilistic detection and tracking of motion discontinuities", *ICCV*, 2:551-558, Greece, 1999.

[10] M. J. Black and A. D. Jepson. "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation", *IJCV*, 26:63-84, 1998.

[11] M. J. Black and A. D. Jepson, "A probabilistic framework for matching temporal trajectories", *ICCV*, 176-181, Greece, 1999.

[12] Y. Boykov and D. Huttenlocher. "Adaptive bayesian recognition in tracking rigid objects", *CVPR*, 697-704, 2000.

[13] S. Boyd and L. Vandenberghe. "Convex optimization", *Cambridge Univ. Press*, 2004.

[14] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. V. Gool. "Robust tracking-by-detection using a detector confidence particle filter", *ICCV*, 2009

[15] Y. Cai, N. Freitas and J. Little. "Robust Visual Tracking for Multiple Targets", *ECCV*, 107-118, 2006.

[16] E. Candès, J. Romberg, and T. Tao. "Stable signal recovery from incomplete and inaccurate measurements", *Comm. on Pure and Applied Math*, 59(8):1207-1223, 2006.

[17] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information",*IEEE Transactions on information theory*, 52(2):489-509, 2006.

[18] E. Candès and T. Tao. "Near optimal signal recovery from random projections: Universal encoding strategies?", *IEEE Trans. on Information Theory*, 52(12):5406-5425, 2006.

[19] E. Candès, and M. Wakin. "An introduction to compressive sampling", *IEEE Signal Processing Magazine*, 25(2):21-30, 2008.

[20] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa. "Compressive Sensing for Background Subtraction", *ECCV*, 2008.

[21] S. Chen, D. Donoho, and M. Saunders. "Atomic decomposition by basis pursuit", *SIAM Rev.*, 43(1):129-159, 2001.

[22] R. T. Collins and Y. Liu. "On-Line Selection of Discriminative Tracking Features", *ICCV*, 346-352, 2003.

[23] R. Collins, X. Zhou, and S. K. Teh. "An Open Source Tracking Testbed and Evaluation Web Site", *PETS*, 2005.

[24] D. Comaniciu, V. Ramesh, and P. Meer. "Kernel-based object tracking", *PAMI*, 25:564-577, 2003.

[25] Compressive sensing resources. *http://dsp.rice.edu/cs*.

[26] D. Conte, P. Foggia, J.-M. Jolion, and M. Vento. "A graph-based, multi-resolution algorithm for tracking objects in presence of occlusions". *Pattern Recognition*, 39(4):562-572, 2006.

[27] *http://www.stanford.edu/˜boyd/l1_ls/*.

[28] W. Dai, M. Sheikh, O. Milenkovic, and R. Baraniuk, "Compressive sensing DNA microarrays", *EURASIP journal on bioinformatics and systems biology*, 2009.

[29] G. Dantzig. Linear Programming and Extensions. *Princeton University Press*, 1963.

[30] M. Davies and N. Mitianoudis. "A simple mixture model for sparse overcomplete ICA", *IEE Proc.-Vision, Image and Signal Processing*, 151(1):35-43, August 2004.

[31] P. Debevec. "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography", *SIGGRAPH*, 189-198, 1998.

[32] D. Donoho. "Neighborly polytopes and sparse solutions of underdetermined linear equations", *Technial Report*, 2005-4, Dept. of Statistics, Stanford Univ., 2005.

[33] D. Donoho. "Compressed Sensing", *IEEE Trans. Inf. Theory*, 52(4):1289-1306, 2006.

[34] D. Donoho. "For most large underdetermined systems of linear equations, the minimal ℓ_1 norm solution is also the sparsest solution", *Communications on Pure and Applied Mathematics*, 59(6):797-829, 2006.

[35] D. L. Donoho, and Y. Tsaig. "Fast Solution of ℓ_1-Norm Minimization Problems When the Solution May Be Sparse", *IEEE trans. on Information Theory*, 54(11):4789-4812, 2008.

[36] A. Doucet, N. de Freitas, and N. Gordon. "Sequential Monte Carlo Methods in Practice". *Springer-Verlag*, 2001, New York.

[37] A. Doucet, S. J. Godsill, and C. Andrieu. "On sequential monte carlo sampling methods for bayesian filtering", *Statistics and Computing*, 10(3):197C209, 2000.

[38] G.J. Edwards, C.J. Taylor, and T.F. Cootes. "Improving Identification Performance by Integrating Evidence from Sequences", *CVPR*, 1:486-491, 1999.

[39] K. Egiazarian, A. Foi, and V. Katkovnik. "Compressed sensing image reconstruction via recursive spatially adaptive filtering", *IEEE Conf. on Image Processing (ICIP)*, San Antonio, Texas, September 2007.

[40] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. "Least angle regression", *Ann. Statist.*, 32(2):407-499, 2004.

[41] C. Fevotte and S. Godsill. “Sparse linear regression in unions of bases via bayesian variable selection”, *IEEE Signal Processing Letters* 13(7):441-444, 2006.

[42] J. Friedman, T. Hastie, H. Hofling and R. Tibshirani. “Pathwise coordinate optimization”, *The Annals of Applied Statistics* vol. 1, 302-332, 2007.

[43] D. Freedman and M. W. Turek. “Illumination-Invariant Tracking via Graph Cuts”, *CVPR*, 10-17, 2005.

[44] L. Gan “Block compressed sensing of natural images”, *Conf. on Digital Signal Processing (DSP)*, Cardiff, UK, July 2007.

[45] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation”, *IEEE Proceedings on Radar and Signal Processing*, (140)107-113, 1993.

[46] V. K. Goyal, M. Vetterli, and N. T. Thao. “Quantized overcomplete expansions in RN: Analysis, synthesis and algorithms”, *IEEE Trans. on Information Theory*, 44,(1):16-31, 1998.

[47] J. Gu, S. Nayar, E. Grinspun, P. Belhumeur, and R. Ramamoorthi. “Compressive Structured Light for Recovering Inhomogeneous Participating Media”, *ECCV*, 2008.

[48] G. Hager and P. Belhumeur. “Real-time tracking of image regions with changes in geometry and illumination”, *CVPR*, 403-410, 1996.

[49] G. D. Hager and P. N. Belhumeur. “Efficient region tracking with parametric models of geometry and illumination”, *IEEE PAMI* 20:1025-1039, 1998.

[50] P. Hallinan. “A Low-dimensional Representation of Human Faces for Arbitrary Lighting Conditions”, *CVPR*, 995-999, 1994.

[51] B. Han, and L. Davis. “On-line density-based appearance modeling for object tracking”, *ICCV*, 1492-1499, 2005.

[52] K. Hara, K. Nishino, and K. Ikeuchi. “Determining reflectance and light position from a single image without distant illumination assumption”, *ICCV*, 1:560-567, 2003.

[53] K. Hara, K. Nishino, and K. Ikeuchi. “Mixture of Spherical Distributions for Single-View Relighting”, *PAMI*, 30(1):25-35, 2008.

[54] R. Hartley, and A. Zisserman. "Multiple View Geometry in Computer Vision", *Cambridge University Press*, March, 2004.

[55] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The entire regularization path for the support vector machine", *J. Mach. Learning Res.*, 1391-1415, 2004.

[56] J. Ho, K.-C. Lee, M.-H. Yang, and D. Kriegman. "Visual tracking using learned subspaces", *CVPR*, 782-789, 2004.

[57] J. Huang, X. Huang, and D. Metaxas. "Simultaneous Image Transformation and Sparse Representation Recovery", *CVPR*, 2008.

[58] J. Huang, X. Huang, and D. Metaxas. "Learning With Dynamic Group Sparsity", *ICCV*, 2009.

[59] Y. Huang and I. A. Essa. "Tracking Multiple Objects through Occlusions". *CVPR* 2:1051-1058, 2005.

[60] M. Isard and A. Blake. "Contour tracking by stochastic propagation of conditional density", *European Conference on Computer Vision*, 343-356, Cambridge, UK, 1996.

[61] M. Isard and A. Blake. "Condensation - conditional density propagation for visual tracking", *IJCV*, 29:5-28, 1998.

[62] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. "Robust online appearance models for visual tracking", *PAMI*, 25:1296-1311, 2003.

[63] S. Ji, Y. Xue, and L. Carin. "Bayesian Compressive Sensing", *IEEE Trans. Signal Processing*, 56(6):2346-2356, 2008.

[64] S. Ji, D. Dunson, and L. Carin. "Multi-Task Compressive Sensing", *IEEE Trans. Signal Processing*, 57(1):92-106, 2009.

[65] C. Johnson, J. Seidel, and A. Sofer. "Interior point methodology for 3-D PET reconstruction", *IEEE Trans. Med. Imag.*, 19(4):271-285, 2000.

[66] R. E. Kalman. "A new approach to linear filtering and prediction problem", *Trans. of the ASME - Journal of Basic Engineering*, 82:35-45, 1960.9, 13.

[67] T. Kaneko and O. Hori. "Feature Selection for Reliable Tracking using Template Matching", *CVPR*, 796-802, 2003.

[68] Z. Khan, T. Balch, and F. Dellaert. "A Rao-Blackwellized particle filter for Eigen-Tracking", *CVPR*, 980-986, 2004.

[69] T. Kim, and K. Hong. "A practical single image based approach for estimating illumination distribution from shadows", *ICCV*, 266-271, 2005.

[70] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. "A method for large-scale ℓ_1-regularized least squares", *IEEE J. on Selected Topics in Signal Processing*, 1(4):606-617, 2007.

[71] K. Kim, and L. Davis. "Multi-camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering", 98-109, ECCV, 2006.

[72] G. Kitagawa. "Monte carlo filter and smoother for non-gaussian nonlinear state space models", *Journal of Computational and Graphical Statistics*, 5:1-25, 1996.

[73] J. Lalonde, A. A. Efros, and S. G. Narasimhan. "Estimating Natural Illumination from a Single Outdoor Image", *ICCV*, 2009.

[74] X. Li, W. Hu, Z. Zhang, and X. Zhang. "Robust visual tracking based on an effective appearance model", *ECCV*, 396-408, 2008.

[75] X. Liu, and T. Chen."Video-Based Face Recognition Using Adaptive Hidden Markov Models", *CVPR*, 1:340-345, 2003.

[76] J. S. Liu and R. Chen. "Sequential monte carlo for dynamic systems", *Journal of the American Statistical Association*, 93:1031-1041, 1998.

[77] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman."Video-Based Face Recognition Using Probabilistic Appearance Manifolds", *CVPR*, 1:313-320, 2003.

[78] K.-C. Lee, and D. Kriegman. "Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking", *CVPR*, 852-859, 2005.

[79] B. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision", *IJCAI*, 674-679, 1981.

[80] M. Lustig, D. Donoho, and J. M. Pauly. "Sparse MRI: The application of compressed sensing for rapid MR imaging", *Magnetic Resonance in Medicine*, 58(6):1182-1195, 2007.

[81] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. "Discriminative learned dictionaries for local image analysis", *CVPR*, 2008.

[82] R. Marcia, and R. Willett. "Comprssive coded aperture video reconstruction", *European Signal Processing Conference*, 2008.

[83] I. Matthews, T. Ishikawa, and S. Baker. "The template update problem", *PAMI*, 810-815, 2004.

[84] X. Mei, S. K. Zhou, and F. Porikli. "Probabilistic Visual Tracking via Robust Template Matching and Incremental Subspace Update". *ICME*, 1818-1821 , 2007.

[85] X. Mei, and F. Porikli. "Fast image registration via joint gradient maximization: application to multi-modal data", *Proceedings of SPIE Volume 6395 Electro-Optical and Infrared Systems: Technology and Applications III*, 63950P.1-63950P.5, 2006.

[86] X. Mei, and F. Porikli. "Joint tracking and video registration by factorial Hidden Markov models", *ICASSP*, 973-976, 2008.

[87] X. Mei, S. K. Zhou and H. Wu. "Integrated Detection, Tracking and Recognition for IR Video-Based Vehicle Classification", *ICASSP*, 745-748, 2006.

[88] X. Mei, S. K. Zhou, H. Wu, and F. Porikli. "Integrated Detection, Tracking and Recognition for IR Video-Based Vehicle Classification", *Journal of Computers*, 2(6), 1-8, 2007.

[89] X. Mei, H. Ling, and D.W. Jacobs. "Sparse Representation of Cast Shadows via ℓ_1-Regularized Least Squares", *ICCV*, 2009.

[90] X. Mei and H. Ling. "Robust visual tracking using ℓ_1 minimization", *ICCV*, 2009.

[91] B. Mercier, D. Meneveaux, and A. Fournier. "A Framework for Automatically Recovering Object Shape, Reflectance and Light Sources from Calibrated Images", *IJCV*, 73(1):77-93, 2007.

[92] J. Mooser, Q. Wang, S. You, and U. Neumann. "Fast Simultaneous Tracking and Recognition Using Incremental Keypoint Matching", *3DPVT*, 2008.

[93] Y. Moses. "Face Recognition: Generatlization to Novel Images", *PhD theis*, Weizmann Institute of Science, 1993.

[94] A. Nag, D.J. Miller, A. P. Brown, and K. J. Sullivan. "A system for vehicle recognition in video based on SIFT features, mixture models, and support vector machines", *SPIE*, 2007.

[95] G. Narkiss, and M. Zibulevsky. “Sequential Subspace Optimization Method for Large-Scale Unconstrained Problems”, The Technion, Haifa, Israel, Tech. Rep. CCIT No 559, 2005.

[96] A. Neumaier.“Solving ill-conditioned and singular linear systems: A tutorial on regularization”, *SIAM Rev.*, 40(3):636666, 1998.

[97] Y. Nesterov and A. Nemirovsky. “Interior-point polynomial methods in convex programming”, *Studies in Applied Mathematics*, 1994.

[98] H. T. Nguyen and A. Smeulders. “Tracking Aspects of the Foreground against the Background”, *ECCV*, 446-456, 2004.

[99] J. Nimeroff, E. Simoncelli, and J. Dorsey. “Efficient Re-rendering of Naturally Illuminated Environments”, *Eurographics Workshop on Rendering*, 1994.

[100] K. Nishino, Z. Zhang, and K. Ikeuchi. “Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis”, *ICCV*, 1:599-606, 2001.

[101] R. Ng, R. Ramamoorth, and P. Hanrahan. “All-Frequency Shadows Using Nonlinear Wavelet Lighting Approximation”, *SIGGRAPH*, 2003.

[102] B. North, A. Blake, M. Isard, and J. Rittscher. “Learning and classification of complex dynamics”, *IEEE PAMI*, 22:1016-1034, 2000.

[103] T. Okabe, I. Sato, and Y. Sato. “Spherical Harmonics vs. Haar Wavelets: Basis for Recovering Illumination from Cast Shadows”, *CVPR*, 50-57, 2004.

[104] M. Osborne, B. Presnell, and B. Turlach. “A new approach to variable selection in least squares problems”, *IMA J. Numer. Anal.*, 20(3):389-403, 2000.

[105] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. “An iterative regularization method for total variation based image restoration”, *SIAM J. Multiscale Modeling and Simulation*, 4(2):460-489, 2005.

[106] A. Panagopoulos, D. Samaras, and N. Paragios. “Robust Shadow and Illumination Estimation Using a Mixture Model”, *CVPR*, 2009.

[107] P. Peers, D. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi, and P. Debevec. “Compressive light transport sensing”, *ACM Trans. on Graphics*, 28(1), 2009.

[108] F. Porikli, O. Tuzel and P. Meer. “Covariance tracking using model update based on lie algebra”, *CVPR*, 728-735, 2006.

[109] J. Provost, and F. Lesage. “The application of compressed sensing for photo-acoustic tomography”, *IEEE Trans Med Imaging*, 28(4):585-94, 2009.

[110] R. Ramamoorthi and P. Hanrahan. “On the relationship between radiance and irradiance: determining the illumination from images of a convex Lambertian object”, *JOSA A*, 10:2448-2459, 2001.

[111] R. Ramamoorthi and P. Hanrahan. “A Signal-Processing Framework for Inverse Rendering”, *SIGGRAPH*, 1:117-128, 2001.

[112] R. Ramamoorthi, M. Koudelka, and P. Belhumeur. “A Fourier Theory for Cast Shadows”, *PAMI*, 24(2):288-295, 2005.

[113] J. Romberg. “Imaging via compressive sampling”, *IEEE Signal Processing Magazine*, 25(2):14-20, 2008.

[114] D. A. Ross, J. Lim, R. Lin and M. Yang. “Incremental learning for robust visual tracking”, *IJCV*, 77:125-141, 2008.

[115] S. Rosset, L. Saul, Y. Weiss, and L. Bottou, “Tracking curved regularized optimization solution paths”, *Advances in Neural Information Processing Systems* Cambridge, MA: MIT Press, 2005.

[116] J. Sakagaito, and T. Wada. “Nearest first traversing graph for simultaneous object tracking and recognition”, *CVPR*, 1-7, 2007.

[117] S. Shirdhonkar and D. Jacobs. “Non-Negative Lighting and Specular Object Recognition”, *ICCV*, 1323-1330, 2005.

[118] I. Sato, Y. Sato, and K. Ikeuchi. “Stability Issues in Recovering Illumination Distribution from Brightness in Shadows”, *CVPR*, 400-407, 2001.

[119] I. Sato, Y. Sato, and K. Ikeuchi. “Illumination Distribution from Shadows”, *CVPR*, 306-312, 1999.

[120] I. Sato, Y. Sato and K. Ikeuchi. “Illumination from Shadows”, *PAMI*, 25(3):290-300, 2003.

[121] P. Sen and S. Darabi. “Compressive Dual Photography”, *Eurographics*, 28(2), 2009.

[122] A. Shashua. "On Photometric Issues in 3d Visual Recognition From a Single 2d Image", *IJCV*, 21(1-2):99-122, 1997.

[123] M. Sheikh, O. Milenkovic, and R. Baraniuk. "Designing compressive sensing DNA microarrays", *IEEE Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, St. Thomas, U.S. Virgin Islands, December 2007.

[124] P. Shirley, C. Wang, and K. Zimmerman. "Monte carlo techniques for direct lighting calculations", *ACM Transactions on Graphics*, 15(1):1-36, 1996.

[125] H. Sidenbladh, M. J. Black, and D. J. Fleet. "Stochastic tracking of 3d human figures using 2d image motion", *European Conference on Computer Vision*, 2:702-718, Copenhagen, Denmark, 2002.

[126] P. Sloan, J. Kautz, and J. Snyder. "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments", *SIGGRAPH*, 2002.

[127] P. Sloan, J. Hall, J. Hart, and J. Snyder. "Clustered principal components for precomputed radiance transfer", *SIGGRAPH*, 382-391, 2003.

[128] J. Snyder, P. Sloan, and Y. Ng. "Systems and methods for all-frequency relighting using spherical harmonics and point-light distributions", US Patent 7,262,771.

[129] D. S. Taubman, and M. W. Marcellin. *JPEG 2000: Image compression fundamentals, standards and practice*, Norwell, MA: Kluwer, 2001.

[130] J. Tropp And A Gilbert. "Signal recovery from random measurements via orthogonal matching pursuit", *IEEE Trans. Inform. Theory*, 53(12): 4655C4666, 2007.

[131] J. Trzasko, A. Manduca, and Eric Borisch, "Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0-minimization", *IEEE Trans. Medical Imaging*, 28(1):106-121, 2009.

[132] Y. Tsai, and Z. Shih. "All-Frequency Precomputed Radiance Transfer using Spherical Radial Basis Functions and Clustered Tensor Approximation", *SIGGRAPH*, 25(3):967-976, 2006.

[133] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian and D. P. Greenberg. "Lightcuts: a scalable approach to illumination", *PAMI*, 24(3):1098-1107, 2005.

[134] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk. "An architecture for compressive imaging". *Int. Conf. on Image Processing (ICIP)*, 1273-1276, 2006.

[135] O. Williams, A. Blake, and R. Cipolla. "Sparse Bayesian Learning for Efficient visual Tracking", *PAMI*, 27:1292-1304, 2005.

[136] S. Wright, "Primal-dual interior-point methods", *Society for Industrial and Applied Mathematics*, 1997.

[137] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. "Robust Face Recognition via Sparse Representation", *PAMI*, 31(1):210-227, 2009.

[138] Y. Wu and T. S. Huang, "A co-inference approach to robust visual tracking", *ICCV*, 2:26-33, Vancouver, BC, 2001.

[139] C. Yang, R. Duraiswami, and L. S. Davis. "Fast multiple object tracking via a hierarchical particle filter", *ICCV*, 212-219, 2005.

[140] M. Yang, Y. Wu, and G, Hua. "Context-Aware visual tracking", *PAMI*, 31(7):1195-1209, 2009.

[141] Y. Ye, *Interior Point Algorithms: Theory and Analysis*, New York:Wiley, 1997.

[142] A. Yilmaz, O. Javed, and M. Shah. 'Object tracking: A survey". *ACM Comput. Surv.* 38(4), 2006.

[143] W. Yin, S Osher, D. Goldfarb, and J. Darbon. "Bregman iterative algorithms for ℓ_1 minimization with applications to compressed sensing", *SIAM J. Imaging Sci*, 143-168, 2007.

[144] Q. Yu, T. B. Dinh and G. Medioni. "Online tracking and reacquistion using co-trained generative and discriminative trackers", *ECCV*, 678-691, 2008.

[145] Y. Zhang and Y. Yang. "Illuminant direction determination for multiple light sources", *CVPR*, 1:269-276, 2000.

[146] X. Zhang, W. Hu, S. Maybank, X. Li, and M. Zhu. "Sequential particle swarm optimization for visual tracking", *CVPR*, 1-8, 2008.

[147] P. Zhao and B. Yu. "ON model selection consistency of lasso", *J. Machine Learning Research*, 2541-2567, 2006.

[148] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video", *Computer Vision and Image Understanding*, 91:214-245, 2003.

[149] S. K. Zhou, R. Chellappa, and B. Moghaddam. “Visual tracking and recognition using appearance-adaptive models in particle filters”, *IEEE Trans. Image Processing*, 11:1491-1506, 2004.

[150] W. Zhou and C. Kambhamettu. “A unified framework for scene illuminant estimation”, *Image and Vision Computing*, 26(3):415-429, 2008.

[151] Y. Zhou and H. Tao. “A background layer model for object tracking through occlusion”, *ICCV*, 2003.