Compositional synthesis of optical imaging systems

Devika Subramanian and Ron Goldman
Department of Computer Science
Rice University
Houston TX 77005
devika@cs.rice.edu, (713) 348-5661

November 2000

Abstract

This paper introduces a new, systematic and complete compositional technique for conceptual design of imaging optical systems from behavioral specifications. There are two key ideas: (1) modeling the structure and behavior of optical components using affine transformations, and (2) using a systematic search algorithm integrated with a powerful algebraic constraint solver to compute three-dimensional layouts of optical components in the design. By embedding our tool in a constraint programming environment, we are able to automate the conceptual design of optical imaging systems — a step performed manually by human designers until now. We demonstrate the power of the method by recreating several designs for imaging systems of copiers.

1 Introduction

The goal of our research is to derive computational theories of conceptual or pre-parametric design for optical imaging systems. Examples of such systems include VCRs, optical gyroscopes, and imaging systems of cameras and copiers. Optical imaging system synthesis solves the problem of determining the three-dimensional structure of a device composed of optical components that implements a given behavioral specification. Optical and optomechanical systems are ubiquitous today; yet there are few programs that can automatically design and simulate them at a conceptual or pre-parametric level.

A typical design cycle for optomechanical systems is shown in Figure 1. The most critical design decisions are made in the pre-parametric design stage: viz., conceptual studies and layout design. Errors made at this level are very costly to correct at the later stages. Conceptual design is difficult because a variety of task-specific constraints ranging from geometry, kinematics, dynamics, materials, assembly, and costs need to be considered *simultaneously*. The standard method for generating conceptual designs is to hold brainstorming sessions with human experts from each of these areas to propose, check, and satisfy constraints from their individual domains. This process is slow, non-systematic, uncoordinated, and error-prone since it leaves no audit trail of key design decisions. It can lead to disasters when people overlook important constraints or when not all feasible alternatives are considered [Pet82].

This paper proposes a design tool for automatic generation of conceptual alternatives, i.e., high level descriptions of the topology of a device. We distinguish between pre-parametric conceptual design and parametric design. Conceptual design generates device configurations and establishes constraints on key design parameters. Parametric design involving finding optimal values of parameters for a given configuration The key idea in our paper is to construct design topologies by decomposing given behavioral specifications into primitive design fragments which can then be implemented using available components. The specification of desired behaviors is in the form of sets of constraints on object and image configurations, and the problem of generating alternate conceptual layouts is formulated as constraint satisfaction. Embedding the design tool in a constraint programming environment enables rapid generation of conceptual alternatives, and testing designs by simulation to verify behavior. Underlying the design algorithms is a theory of compositional synthesis grounded in the following facts: (1) the behavior of optical components can be expressed as affine transformations on object and image configurations, (2) the overall input-output behavioral specification of the system itself can be expressed as an affine transformation on object and image configurations, and (3) a powerful algebraic constraint solver can systematically compose component behaviors *symbolically* and check if they implement the desired system behavior.

We illustrate our design system with an example. Consider the synthesis of an imaging system for a photocopier, where the task is to project an image of a stationary document page on to an imaging drum. To specify this design problem, we describe the position, orientation and scale of the image with reference to the object. We treat the image as a scaled, rotated, and translated version of the object.

$$
\begin{array}{ll}
Input: & \left\{ \begin{array}{ll} \text{obj pos} & \text{(0,0,0)} \\ \text{obj orient} & \text{((1,0,0),(0,1,0),(0,0,1))} \end{array} \right. \\
Output: & \left\{ \begin{array}{ll} \text{img pos} & \text{(1,1,0)} \\ \text{img orient} & \text{((-1,0,0),(0,-1,0),(0,0,1))} \\ \text{img scale} & \text{((0,0,0),((1,(1,0,0)),(1,(0,1,0)),(1,(0,0,1))))} \\ \text{real?} & + \end{array} \right.
\end{array}
$$

This is a partial specification of the device that captures its optical input-output behavior. The object is at the origin; the real image which is the same size as the object (unity scaling along x, y and z axes with scaling origin $(0,0,0)$) is at (1,1,0) rotated by π about the z axis with respect to the object orientation. We can also specify constraints on the volume or area of the overall device, or on the total optical path length. For this example, however, we focus on the input-output transformation alone, and compute a set of layouts of the optical components that achieves the given behavior.

The design generated by our system, consisting of a single lens and a pair of mirrors, is shown in Figure 2. In fact, our program outputs a family of designs parameterized by x, which is the shared x-coordinate of the mirrors and lens. The parameter x is in the open interval $(0,1)$. This is the smallest design in the sense of the fewest number of components needed to achieve the optical behavior. This design family has the principal lens axis parallel to the y axis. When we orient the principal axis of the lens to be parallel to $(-1/\sqrt{2}, 1/\sqrt{2}, 0)$, we obtain the design shown in Figure 3.

There are four unique aspects of our method. First, our layout synthesis algorithm is grounded in the laws of optics with optical components represented as linear transformations on object and image configurations. Second, our algorithm is complete; if enumerates all feasible design families

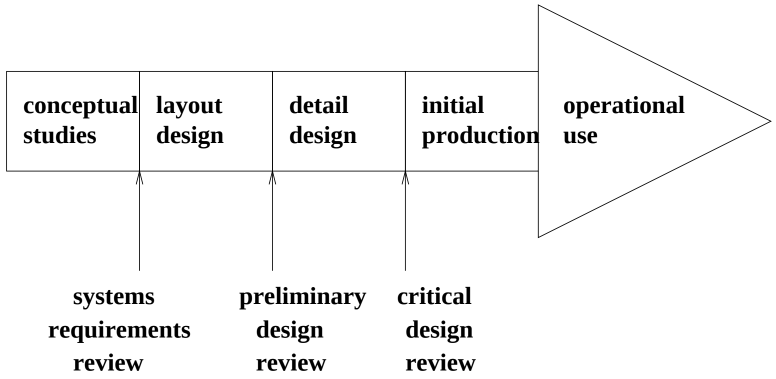

Figure 1: A typical design review cycle for optomechanical systems. Adapted from P. R. Yoder Jr.'s text *Optomechanical System Design*, Marcel Dekker Inc., 1986.

that implement a given specification. Since our algorithm explores designs in increasing order of complexity as specified by the number of primitive components, we can guarantee minimality for the designs produced. For this example specification, there are no other designs involving fewer mirrors and lenses. Third, our algorithm computes key constraints on positions and orientations for all components in the design. These can then be instantiated further for input into a detailed design program for further optimization and analysis. Our approach essentially allows for rapid generation of conceptual design sketches that can be refined with further computation. Fourth, even though we work from first principles in optics, we can obtain real-time generation of alternative design topologies. Our synthesis algorithm is fully implemented in *Mathematica*, and all of the examples presented in this paper are the output of our design system rendered using the Optica application in *Mathematica*. Our system has produced interesting designs for imaging systems for copiers as shown in Section 5.

This paper is organized as follows. In Section 2, we define object and image configurations and introduce transformations that map object to image configurations. We represent input-output specifications of optical imaging systems as affine transformations, viz., products of translation, rotation and scaling operations. We formulate the behavior of devices like mirrors, lenses and prisms (treated as special mirrors) as affine transformations. The operations that compose specifications and their concrete implementations are introduced at the end of Section 2. The foundations of a compositional theory of synthesis are described in Section 3. The general algebraic synthesis

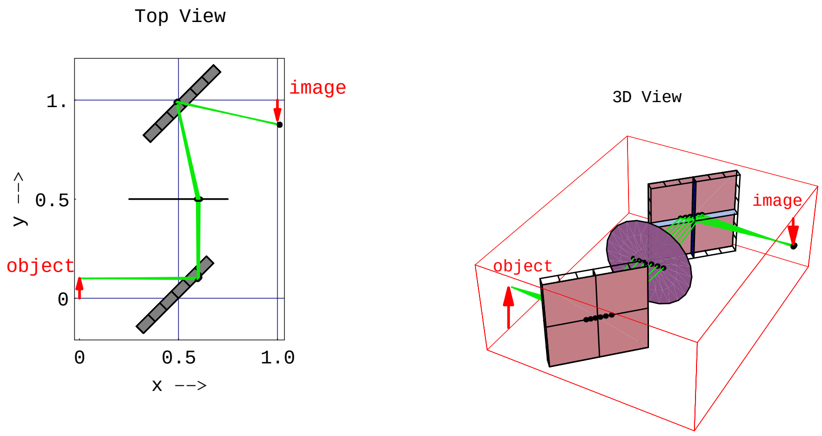

Figure 2: An imaging system designed by our synthesis program. The top view and 3-D arrangement of components are shown. The system transforms an object at (0,0,0) into a real image of the same size at (1,1,0) rotated by π about the z axis with respect to the object.

technique based on symbolic generation and testing of compositions of optical components is presented in Section 4. We demonstrate that the problem of designing layouts from specifications of object and image configurations is under-constrained. To overcome this problem, we develop a systematic search algorithm that refines problem specifications by adding three generic constraints including one on the orientation of the principal optical axis of the system. In Section 5, we present computational results obtained with our framework in the domain of imaging systems for copiers. We conclude with a discussion of related work and avenues for future research in Section 6.

2 Specifying the behavior of optical imaging systems

We specify optical systems by their input-output optical behavior: i.e., the position, orientation and scale of the image produced by the system, with respect to an object. A design is a sequence of optical components with associated position and orientation information, that implements the given specification.

Conceptual synthesis of optical imaging systems is generally acknowl-

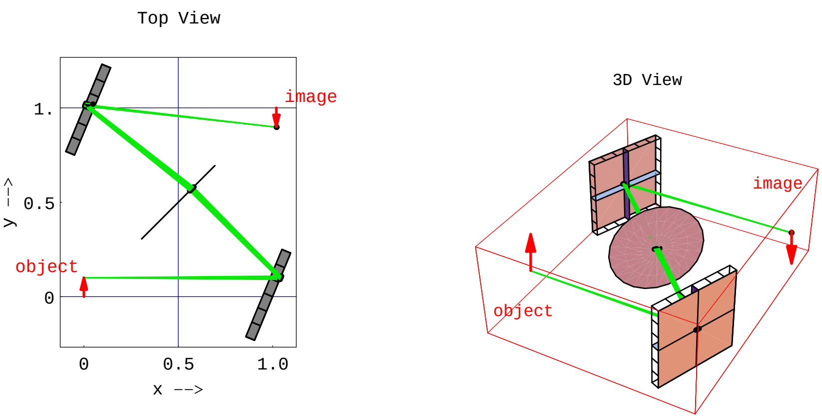

Figure 3: Another imaging system designed by our synthesis program for the problem described in Figure 1. Note that the only difference between the two designs is the orientation of the principal axis of the lens.

edged to be a difficult problem because the input-output specifications alone yield a severely under-constrained system of equations. Constraints on orientation of principal optical axis of the system, or optical path length generally need to be added to constrain the space of possible designs. Further, even with these constraints in place, there are usually an infinite number of possible designs that satisfy the given specifications. So a synthesis technique has to compactly represent a parametric family of designs, rather than a single design.

While there is a rich theory for ray propagation through optical systems [GB75], the theory is not readily invertible for use as a generator of designs. There is no compositional theory of design of such systems which would make it possible to exhaustively and systematically enumerate *all* systems that achieve a given optical input-output specification. Our paper contributes to this area by providing a mathematical foundation for compositional design of optical systems, and by implementing a new, practical, systematic design algorithm for imaging system synthesis from behavior specifications.

2.1 Specifying optical behavior using affine transformations

We specify optical behavior in terms of configurations of objects and images. Objects and images are rigid bodies in $\Re^3$ with positions and orientations. Images can be real or virtual. Mirrors construct virtual images, while astronomical telescopes construct real images. Real images form a visible

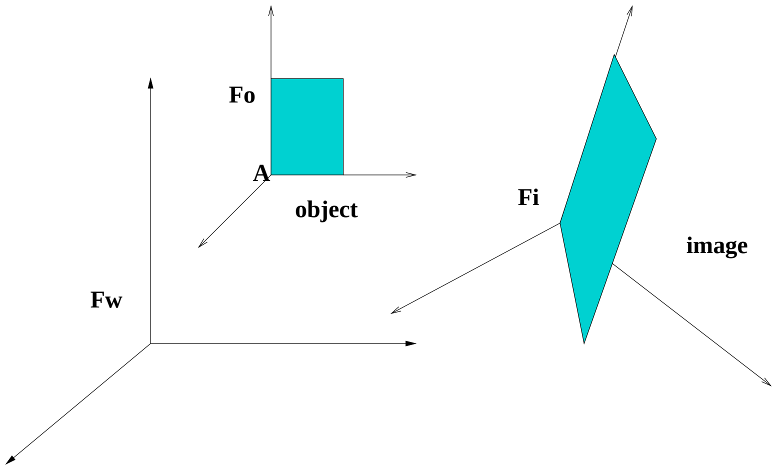

Figure 4: Object and image configurations.

projection on a screen while virtual images do not. Copiers and cameras need to form real images of pages and pictures onto their imaging surfaces.

We specify the configuration of objects and images with three items of information: their position and orientation with respect to a fixed frame of reference, and a boolean indicating whether they are real or virtual. By definition, objects are real.

Let F_o be a Cartesian frame embedded in an object and let F_W be a fixed frame of reference. The origin of F_o is a reference point on the object (see Figure 4).

Definition 1 *A configuration of A is a specification of the position and orientation of the local coordinate frame F_A with respect to a fixed frame of reference F_W, together with $r \in \{-,+\}$ indicating whether the configuration is virtual or real. The configuration space of A is the set of all possible configurations of A.*

While the configuration space $\mathcal{C}$ of A is intrinsically independent of the choice of coordinate frames F_A and F_W, its representation depends on these two choices. The configuration space of an object at the origin of a fixed Cartesian reference frame and oriented parallel with it is $\{(0,0,0),((1,0,0),(0,1,0),(0,0,1)),+)\}$. The fact that it is real is captured in the third component of the configuration, which is a $+$.

An optical transformation is a relation $\mathcal{R} \subseteq O \times I$ where O and I are the configuration spaces of the object and image respectively. $\mathcal{R}$ is a connected subset of $O \times I$. We focus on relations between object and image configurations that can be expressed as some composition of translation, rotation and scaling transformations on the object configuration, since these relations constitute all possible specifications in Gaussian optics.

For example, from the point of view of a Cartesian frame attached to the object, the image coordinate frame in the example in Figure 2 is generated by rotation by π about the z axis of the object frame at $(0,0,0)$, and by translation of the frame by $(1,1,0)$. Let $R((0,0,0),(0,0,1),\pi)$ denote the rotation transformation and $T((1,1,0))$ denote the translation. Then, the relation $\mathcal{R}$ between object and image configurations is the function f which is the composition (denoted by "*") of the rotational and translational transformations. o_P and i_P are the origins of the object and image coordinate systems o_C and i_C respectively.

$$\begin{aligned}\mathcal{R} &= \{(o_P, o_C, +), (i_P, i_C, +) | i_P = f(o_P), i_C = f(o_C), \\ &\quad f = R((0,0,0),(0,0,1),\pi) \; * \; T((1,1,0))\}\end{aligned}$$

The imaging system of a simple Galilean telescope with its optical axis oriented along the z axis with respect to F_W can be specified as follows. The specification indicates that a virtual upright image is formed by the telescope. The image is non-uniformly scaled by a factor of m^2 along the x axis, and m along the y and z axes. It is translated by e along the z axis with respect to the object frame.

$$\begin{aligned}\mathcal{R}_{Galilean} &= \{(o_P, o_C, +), (i_P, i_C, -) | i_P = f(o_P), i_C = f(o_C), \\ &\quad f = S((0,0,0),(m^2,(1,0,0)),(m,(0,1,0)),(m,(0,0,1))) \; * \\ &\qquad T((0,0,e))\}\end{aligned}$$

2.2 Affine transformations

Translations, scaling and rotations belong to a family of transformations called *affine transformations.*

Definition 2 *(Affine Transformation) A transformation $\mathcal{T}$ from $\Re^3 \rightarrow \Re^3$ is called affine if $T(u+v) = T(u) + T(v)$, for every vector pair $u, v \in \Re^3$, $T(cv) = cT(v)$ for all constants $c \in \Re$ and vector $v \in \Re^3$, and $T(P+u) = T(P) + T(u)$, for every vector u and every point P in $\Re^3$.*

Informally, any non-singular transformation preserving collinearity and ratios of distances is affine — i.e., all points on a line initially, stay on a line after the transformation, and the midpoint of a line segment remains the midpoint after the transformation. If $\mathcal{T}$ is an affine transformation, then $\mathcal{T}$ is a continuous function. Translations, rotations, and uniform and non-uniform scalings, are all affine transformations.

Affine transformations can be represented by 4×4 matrices in affine coordinates [Gol94]. As examples, we show the matrix representations of translation, rotation and uniform as well as non-uniform scaling.

The matrix representation of translation by a vector $w \in \Re^3$ is shown below where I is a 3×3 identity matrix, and $\mathbf{0}$ is a 3×1 array of zeros.

$$\mathcal{T}(w) = \begin{bmatrix} I & \mathbf{0} \\ w & 1 \end{bmatrix}$$

To translate a point $P = (x, y, z) \in \Re^3$ by w, we multiply $\mathcal{T}(w)$ by the representation of P in affine coordinates, viz., $(x, y, z, 1)$ yielding the expected conclusion that P is translated to $P + w$.

$$\begin{aligned} (P,1) * \mathcal{T}(w) &= (P,1) * \begin{bmatrix} I & \mathbf{0} \\ w & 1 \end{bmatrix} \\ &= (P * I + w, 1) \\ &= (P + w, 1) \end{aligned}$$

Non-uniform scaling about scaling origin Q along vector w by scaling factor c has the following matrix representation.

$$\mathcal{NS}(Q, c, w) = \begin{bmatrix} I - (1-c)(w^T * w) & \mathbf{0} \\ (1-c)(Q.w)w & 1 \end{bmatrix}$$

Uniform scaling about point Q with scale factor c is represented as

$$\mathcal{US}(Q, c) = \begin{bmatrix} cI & \mathbf{0} \\ (1-c)Q & 1 \end{bmatrix}$$

Rotation about point Q and vector w by angle θ is represented as

$$\mathcal{R}(w, Q, \theta) = \begin{bmatrix} R(w,\theta) & \mathbf{0} \\ Q - Q * R(w,\theta) & 1 \end{bmatrix}$$

where $w = (w_1, w_2, w_3)$, $R(w,\theta) = cos(\theta)I + (1 - cos(\theta))(w^T * w) + sin(\theta)w\times$, and

$$w\times = \begin{bmatrix} 0 & w_3 & -w_2 \\ -w_3 & 0 & w_1 \\ w_2 & -w_1 & 0 \end{bmatrix}$$

Optical transformations are implemented by components like mirrors, lenses and prisms. We specify optical components using affine transformations. While [Nus98] and [GB75, BP91] model ray propagation through optical systems and treat optical components as transformers of ray heights and ray angles, we model optical components and optical behavior as transformations on object and image configurations as in [Pen96].[1] The ability to represent optical behavior in terms of affine transformations on object configurations allows us to develop a compositional theory of optical layout synthesis that cleanly integrates conceptual design with the generation of key numerical parameters for the design.

2.3 Modeling optical components

Optical components implement particular affine transformations. We use Gaussian optics, i.e., the optics of paraxial imaging, to model the behavior of optical components [Nus98].[2] As an example, consider the mirror transformation $\mathcal{M}$ — an optical transformation that maps an object to its virtual mirror image. $\mathcal{M}$ is parameterized by the location Q and unit normal n of the mirror. The matrix representation of the mirror transform $\mathcal{M}(n, Q)$ is:

$$\mathcal{M}(n, Q) = \begin{bmatrix} I - 2(n^T * n) & \mathbf{0} \\ 2(Q.n)n & 1 \end{bmatrix}$$

The constraint representation [Gol94] of the mirror transformation $\mathcal{M}(n, Q)$, is
$\mathcal{M}(Q, n) =$

$$\left\{ ((o_P, o_C, +), (i_P, i_C, -)) \;\middle|\; \begin{array}{l} i_C = o_C - 2(o_C.n)n \\ i_P = o_P - 2((o_P - Q).n)n \end{array} \right\}$$

The optical mirror transforms an object consistent with the above equality constraints.[3] No scaling occurs here, the object and the image are of the same size.

[1] [Pen96] uses affine transforms to perform behavioral sensitivity analyses for given designs.

[2] A paraxial ray is a ray that is close enough to the optical axis and whose angular deviation is small enough that sines and tangents of the angles can be replaced by the angles themselves. The paraxial approximation guarantees that rays starting from a given object point intersect at an image point after passing through the optical system.

[3] In our design program, we also add the constraint that $(o_P - Q).n < 0$ to accommodate the fact that the mirror normal faces away from the silvered side of the mirror.

These equations can be represented directly in any constraint programming language and can be used to solve for n and Q given i_P, o_P, i_C, and o_C. In *Mathematica*, which is the constraint programming language we use, we write these equations as follows.

```
Solve[{Ic == Oc - 2 (Oc . Transpose[n]) . n,
       Ip == Op - 2((Op - Q) . Transpose[n]) . n},
      {n,Q}]
```

We can easily see that the constraint and matrix representations are equivalent. A vector $u \in \Re^3$, represented as $(u, 0)$ in affine coordinates, is transformed into $(v, 0)$ as follows. Rewriting $n^T * n$ as $n^T * n$, where n^T is the transpose of n,

$$\begin{aligned}
(v, 0) &= (u, 0) * \mathcal{M}(n, Q) \\
v &= u * (I - 2(n^T * n)) \\
&= u - u * 2(n^T * n) \\
&= u - 2(u * n^T) * n \\
&= u - 2(u.n)n
\end{aligned}$$

A point $P \in \Re^3$, represented as $(P, 1)$ in affine coordinates, is transformed into $(P', 1)$ where

$$\begin{aligned}
(P', 1) &= (P, 1) * \mathcal{M}(n, Q) \\
P' &= P * (I - 2(n^T * n)) + 2(Q.n)n \\
&= P - P * 2(n^T * n) + 2(Q.n)n \\
&= P - 2(P * n^T) * n + 2(Q.n)n \\
&= P - 2(P.n)n + 2(Q.n)n \\
&= P - 2((P - Q).n)n
\end{aligned}$$

The mirror transformation can be implemented with a plane mirror located at Q with its normal parallel to n. It can also be implemented by a suitably positioned prism.

Another important class of optical transformations are scaling transformations which are implemented by lenses. An ideal Gaussian single lens with coordinate frame l_C, position Q, and focal length F, maps an object at position o_P with orientation o_C into an image at i_P with orientation i_C by the composition of a non-uniform scaling, rotation and translation transformation. The coordinate frame l_C associated with the lens is the Cartesian

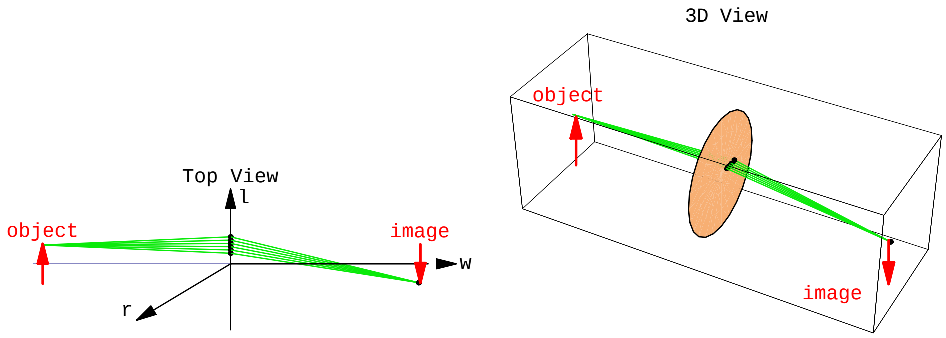

Figure 5: A simple convex lens with focal length F.

frame (w, l, r) where w is the vector aligned with the principal axis, r is the vector about which the lens rotates the image, and l is perpendicular to w and r. The lens coordinate frame has to be aligned with the object frame o_C and the origin of the object frame o_P must lie on the principal axis of the lens. If the object is within the focal length of the lens, a virtual upright image is formed, whereas if the object distance is greater than the focal length, an inverted real image is formed. The magnification factor of the image is the ratio of the image to the object distance. The image is non-uniformly scaled. All of these transforms (rotation, scaling and translation) are governed by the well-known lens law: the sum of the inverses of the image and object distances is the inverse of the focal length of the lens.

Consider the single lens shown in Figure 5 in which we have an object at $o_P = (0, 0, 0)$ with orientation $o_C = ((1, 0, 0), (0, 1, 0), (0, 0, 1))$ at a distance of 20 units from an ideal Gaussian lens with focal length f of 15. The principal lens axis for our example is the pair composed of the vector $w = (1, 0, 0)$ and the lens origin $Q = (20, 0, 0)$. If the image formed by the lens is real, it is rotated by π around the vector $r = (0, 0, 1)$ at the point where it is formed. We have $l = (0, 1, 0)$. The object distance u (resp. image distance v) is the projection of the vector difference between the object position (resp. image) and the lens origin Q projected along the lens axis vector w.

$$\begin{aligned} u &= (Q - o_P) * w^T \\ &= ((20, 0, 0) - (0, 0, 0)) * (1, 0, 0)^T \\ &= 20 \end{aligned}$$

Note the projection of $(Q - o_P)$ on to l and r are zero because $l_C = o_C$. The lens law $\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$ allows us to derive the image distance v and the magnification factor m.

$$\begin{aligned}
\frac{1}{f} &= \frac{1}{u} + \frac{1}{v} \\
\frac{1}{15} &= \frac{1}{20} + \frac{1}{v} \\
v &= 60 \\
v &= (i_P - Q) * w^T \\
i_P &= Q + v * w = (80, 0, 0) \\
m &= \frac{v}{u} = 3
\end{aligned}$$

The image origin i_P also lies on the principal axis of the lens, and the projection of $(i_P - Q)$ onto the vectors r and l are also zero.

The lens transform is the product of the following transformations: a translation, three non-uniform scaling transforms along w, l and r, and a rotation by π about r and i_P since we have an inverted real image. The magnification factor is positive for real images and negative for virtual ones.

$$\begin{aligned}
&\mathcal{T}(i_P - o_P) \\
&\mathcal{NS}(i_P, m, (1,0,0)) \\
&\mathcal{NS}(i_P, m, (0,1,0)) \\
&\mathcal{NS}(i_P, m^2, (0,0,1)) \\
&\mathcal{R}(i_P, (0,0,1), \pi)
\end{aligned}$$

Generalizing, we have the following transformation implemented by a Gaussian lens at position Q with coordinate frame (w, l, r) and focal length F.

$\mathcal{L}(Q, (w,l,r), F, o_P) =$

$$\left\{ ((o_P, o_C, +), (i_P, i_C, sgn(m))) \;\middle\|\; \begin{array}{l}
l_C = o_C \\
i_C = f(o_C) \\
i_P = f(o_P) \\
\frac{1}{F} = \frac{1}{(Q - o_P) * w^T} + \frac{1}{(i_P - Q) * w^T} \\
m = \frac{(i_P - Q) * w^T}{(Q - o_P) * w^T} \\
\theta = \pi \;\text{ if } m > 0 \;\text{ and 0 otherwise} \\
f = \mathcal{T}(i_P - o_P) * \mathcal{NS}(i_P, w, m) * \mathcal{NS}(i_P, l, m) * \\
\quad \mathcal{NS}(i_P, r, m^2) * R(r, i_P, \theta)
\end{array} \right\}$$

o_P and i_P are constrained to lie on the axis defined by w and Q. This means that

$$(Q - o_P) * l^T = 0$$

$$
\begin{aligned}
(Q - o_P) * r^T &= 0 \\
(i_P - Q) * l^T &= 0 \\
(i_P - Q) * r^T &= 0
\end{aligned}
$$

The image is scaled by a factor m along axes w and l and m^2 along r. Note that we only model focal lengths for lenses, and we do not take their radii or thickness into account. We leave this to the detailed design phase for which there are several available programs (e.g., OSLO[OSL94], Winlens [Sch99], Moore[Moo99]).

2.4 Composition of optical transformations

Complex optical systems are constructed by composing simpler ones. The composition of two optical transformations $\mathcal{R}_1$ over $O_1 \times I_1$ and $\mathcal{R}_2$ over $O_2 \times I_2$ is a new optical transformation over $O_1 \times I_2$ defined as

$$\mathcal{R}_1 \bowtie \mathcal{R}_2 = \{(o_1, i_2) | \exists i_1, i_2 \;\; i_1 = o_2 \wedge (o_1, i_1) \in \mathcal{R}_1 \wedge (o_2, i_2) \in \mathcal{R}_2\}$$

Two configuration spaces are equal if and only if they denote the same set of configurations. To impose the equality constraint between two configuration spaces i_1 and o_2, we enforce identity in the real/virtual dimension, and then intersect their positions and orientations to obtain configurations common to both spaces. To implement this composition concretely, we let the image produced by the optical subsystem that implements $\mathcal{R}_1$ serve as the object for the optical subsystem that implements $\mathcal{R}_2$.

We can use the matrix representation of optical behavior to define an equivalent composition operator.

Definition 3 *The composition of two optical transformations $\mathcal{R}_1$ and $\mathcal{R}_2$ is the product of the 4×4 matrices representing each transformation.*

We illustrate the use of the composition definitions to compute the behavior of the system shown in Figure 6. The system is composed of two mirror transformations $\mathcal{M}_1((0,0,0),(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0))$ and $\mathcal{M}_2((0,1,0),(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0))$ defined below. For this example,

$$\mathcal{M}_1((0,0,0),(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)) \;\; = \;\; \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

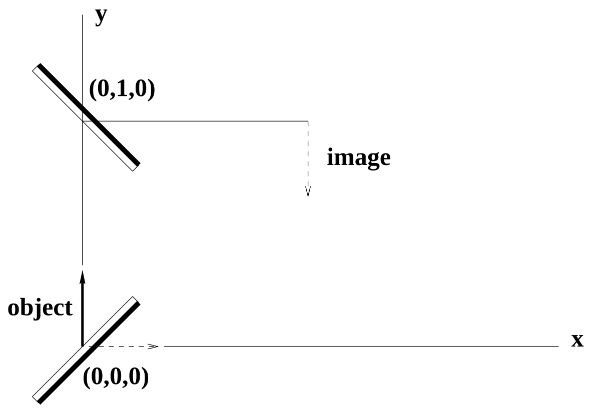

Figure 6: The two mirrors shown above transform an object at the origin oriented along the positive y axis into a virtual image at (1,1,0) oriented along the negative y axis. The silvered side of the mirror is indicated with a thick line.

$$\mathcal{M}_2((0,1,0),(-\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}},0)) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Their product is

$$\mathcal{M}_1 * \mathcal{M}_2 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

We can see that

$$\begin{aligned} (\alpha,\beta,\gamma,0) * \mathcal{M}_1 * \mathcal{M}_2 &= (-\alpha,-\beta,\gamma,0) \\ (x,y,z,1) * \mathcal{M}_1 * \mathcal{M}_2 &= (1-x,1-y,z,1) \end{aligned}$$

Here is another example of the use of the composition definition. Consider the three mirror transformation called a corner cube in [O'S85]. The concrete implementation of the composite transformation is with mirrors that form three sides of a cube. The normals of each of these mirrors parallel the x, y and z axes. The mirrors are positioned at $(0,1,1)$, $(1,0,1)$ and

$(1,1,0)$ respectively. To determine the transformation achieved by the corner cube, we compute the composition $\mathcal{M}_1((0,0,1),(1,1,0)) \bowtie \mathcal{M}_2((0,1,0),(1,0,1)) \bowtie \mathcal{M}_3((1,0,0),(0,1,1))$, which is:

$$\mathcal{M}_1 * \mathcal{M}_2 * \mathcal{M}_3 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It is easy to see that

$$(\alpha,\beta,\gamma,0) * \mathcal{M}_1 * \mathcal{M}_2 * \mathcal{M}_3 = (-\alpha,-\beta,-\gamma,0)$$

The corner cube rotates a vector by π along an axis perpendicular to the vector. This derivation shows the singular property of a corner cube reflector, viz., a light ray incident upon it is returned parallel to itself after three reflections. Arrays of corner cube reflectors were left on the moon during three of the Apollo missions so that light pulses from the earth could be timed to determine the precise distance from the earth to the moon. Because corner cube reflectors were used, the angular orientation of the cube as it was placed on the moon was not critical.

With the paraxial approximation, we model the behavior of plane mirrors and prisms by the mirror image transform $\mathcal{M}(n,Q)$ and that of lenses by the lens transform $\mathcal{L}(Q,(w,l,r),F,o_P)$. Both transforms are particular affine transformations: the mirror accomplishes a rotation, and the lens achieves a composition of scaling, rotation and translation operations. We express the specification of imaging systems also as affine transformations from an object to an image configuration. We will use this formulation in the following section to develop a systematic synthesis procedure for enumerating all topologies that can achieve a given object to image transformation.

3 The Synthesis Problem

We can now cast the problem of synthesizing optical systems from behavior specifications using the framework of affine transformations and their composition introduced in the previous section.

The conceptual synthesis problem is: given an affine transformation M representing the specification of the input-output relation between the object and image configuration spaces of an imaging system, find *all* sequences of optical component transformations $\mathcal{T}_1(a_1),\mathcal{T}_2(a_2),\ldots,\mathcal{T}_n(a_n)$ and instantiations of their associated parameters $a_1,a_2,\ldots,a_n$ such that $\mathcal{T}_1(a_1) \bowtie \ldots \bowtie \mathcal{T}_n(a_n) = M$.

For the copier example from Section 1, we can derive the specification matrix M as follows. The image frame i_C is rotated by π with respect to the z axis of the object frame o_C. The image is unscaled and the image frame is translated by vector $t = i_P - o_P$, with respect to the object frame.

The rotation, scaling and translation matrices from the specification are:

$$\begin{aligned}
R((0,0,1), o_P, \pi) &= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
S(o_P, ((1,(1,0,0)), (1,(0,1,0)), (1,(0,0,1))) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
T(i_P - o_P) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Their product is:

$$R * S * T = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

This is the desired system matrix; it is the product of the rotation, scaling, and translation matrices. We need to synthesize it from a set of primitive optical components. To determine if a composition of primitive optical transformations implements a given specification, we check that the matrix product of the individual primitive transformations equals the system matrix derived from the problem specification.

We will verify that the design proposed in Section 1 does meet the above specification. The design is composed of three primitive optical transforms: a mirror m_1 at $(0.5, 0, 0)$ with normal $(-1/\sqrt{2}, 1/\sqrt{2}, 0)$, a lens l at $(0.5, 0.5, 0)$ with coordinate frame $((0,1,0), (1,0,0), (0,0,1))$ and focal length 0.5, and another mirror m_2 at $(0.5, 1, 0)$ oriented along $(1/\sqrt{2}, -1/\sqrt{2}, 0)$.

$$\begin{aligned}
m_1 &= \mathcal{M}((-1/\sqrt{2}, 1/\sqrt{2}, 0), (0.5, 0, 0)) \\
l &= \mathcal{L}(((0,1,0), (1,0,0), (0,0,-1)), (0.5, 0.5, 0), 0.5, o_P * m_1) \\
m_2 &= \mathcal{M}((1/\sqrt{2}, -1/\sqrt{2}, 0), (0.5, 1, 0))
\end{aligned}$$

The matrix representations of these three transforms are:

$$m1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.5 & -0.5 & 0 & 1 \end{bmatrix}$$

$$l = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

$$m2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.5 & 0.5 & 0 & 1 \end{bmatrix}$$

Their product is

$$m1 * l * m2 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

which is identical to the system matrix. We can conclude that the proposed design does implement the given specification.

Our design space consists of finite compositions of elements from a parametric set of primitive optical transformations, viz., plane mirrors, and lenses. Before we present algorithms for computing all possible compositions that implement a given system specification, we first establish that mirrors and lenses form a *complete* set of primitives. That is, they are sufficient to implement any affine transformation from object to image configuration space. Intuitively, the reason why mirrors and lenses form a complete set of primitives is that mirrors accomplish arbitrary rotations, and lenses produce real images and achieve products of rotations and non-uniform scaling. By positioning them appropriately, we can also realize arbitrary translations. We provide formal proofs of these observations below.

First, we show that mirrors can accomplish arbitrary rotations. For this, we will focus on the upper 3×3 matrix of the mirror transform, i.e., the matrix $I - 2(n^T * n)$. First, note that this matrix is symmetric and orthogonal, i.e., it is identical to its transpose, and that its transpose is also its inverse.

$$(I - 2(n^T * n))^T = I^T - 2(n^T * n)^T$$

$$
\begin{aligned}
&= I - 2(n^T * n)^T \\
&= I - 2(n^T * n)
\end{aligned}
$$

$$
\begin{aligned}
(I - 2(n^T * n)) * (I - 2(n^T * n)) &= (I - 2n^T * n) * (I - 2n^T * n) \\
&= I - 4n^T * n + 4n^T * (n.n) * n \\
&= I - 4n^T * n + 4n^T * n \\
&= I
\end{aligned}
$$

The determinant of $I - 2(n^T * n)$ matrix is which is -1, since $(n^T * n) = 1$ for a normal vector n.

Given an arbitrary 3×3 orthogonal matrix L representing a rotation, we can implement it with no more than a pair of mirror transforms. If the determinant of L is -1, then L can be implemented by a single mirror transform whose normal equals the unit eigenvector of L corresponding to eigenvalue -1[Gol91]. If the determinant of L is $+1$, we can write L as a product of two mirror transforms as shown below. Let L correspond to a rotation of θ about vector $w \in \Re^3$ and point $Q \in \Re^3$. Then,

Proposition 1 *Two mirror transformations $\mathcal{M}_1(n_1, Q_1)$ and $\mathcal{M}_2(n_2, Q_2)$ can generate a rotation of an angle θ about vector w in $\Re^3$ and point $Q \in \Re^3$. In particular, w is parallel to the line that is the intersection of the planes perpendicular to the normals n_1 and n_2, Q is a point on the line to which w is parallel, and the angle ψ between these two planes equals $\theta/2$.*

This proposition corresponds to the well known fact [Lev68] that two mirrors suffice to rotate a vector in $\Re^3$ by some angle θ about an arbitrary vector w and point Q. Here is an elegant derivation of this fact using matrix representations of the mirror and rotation transformations. Let $\mathcal{M}_1(n_1, Q_1)$ and $\mathcal{M}_2(n_2, Q_2)$ be the two mirror transforms; their product is:

$$
\begin{aligned}
\mathcal{M}_1 \bowtie \mathcal{M}_2 &= (I - 2({n_1}^T * n_1)) * (I - 2({n_2}^T * n_2)) \\
&= I - 2({n_1}^T * n_1) - 2({n_2}^T * n_2) + 4({n_1}^T * n_1) * ({n_2}^T * n_2) \\
&= I - 2({n_1}^T * n_1) - 2({n_2}^T * n_2) + 4({n_1}^T * (n_1.n_2) * n_2) \\
&= I - 2({n_1}^T * n_1) - 2({n_2}^T * n_2) + 4(n_1.n_2)({n_1}^T * n_2) \\
&= I - 2({n_1}^T * n_1) - 2({n_2}^T * n_2) + 4cos(\psi)({n_1}^T * n_2)
\end{aligned}
$$

From [Gol91], we can extract both the angle of rotation θ as well as the axis w from the matrix $\mathcal{M} = \mathcal{M}_1 \bowtie \mathcal{M}_2$.

$$
cos(\theta) = \frac{Trace(\mathcal{M}) - 1}{2}
$$

$$
\begin{aligned}
&= \frac{3-2-2+4cos^2(\psi)-1}{2} \\
&= 2cos^2(\psi)-1 \\
cos(\psi) &= \sqrt{\frac{1+cos(\theta)}{2}} \\
\psi &= \frac{\theta}{2}
\end{aligned}
$$

The axis w is a unit eigenvector of $\mathcal{M}$ corresponding to the eigenvalue $+1$.

$$
\begin{aligned}
w &= w*(I-2({n_1}^T*n_1)-2({n_2}^T*n_2)+4cos(\psi)({n_1}^T*n_2)) \\
0 &= -2w*({n_1}^T*n_1)-2w*({n_2}^T*n_2)+4cos(\psi)w*({n_1}^T*n_2) \\
&= -2(w.n_1)*n_1-2(w.n_2)*n_2+4cos(\psi)(w.n_1)*n_2
\end{aligned}
$$

From this we can infer that $w.n_1 = 0$ and $w.n_2 = 0$, which means that w lies on the plane of both mirror transforms; physically w is parallel to the line along which the mirror planes intersect.

The design import of Proposition 1 is that a rotation by a specific angle around a given axis in $\Re^3$ can be accomplished by a single pair of mirrors. However if the goal is to implement the relation $\mathcal{R} = \{(o_P, o_C, +), (i_P, i_C, -) \mid i_C = -o_C\}$ more mirrors may be needed. We can show, for instance, that two mirrors cannot achieve this specification. We require that the product of the rotations produced by the two mirrors with normals n_1 and n_2 equal $-I$. We have the additional constraint that n_1 and n_2 are normal vectors

$$
\begin{aligned}
-I &= (I-2({n_1}^T*n_1))*(I-2({n_2}^T*n_2)) \\
n_1*n_1^T &= 1 \\
n_2*n_2^T &= 1
\end{aligned}
$$

There is no solution to these three equations. The corner cube configuration of three mirrors derived in Section 2 is the smallest design with mirrors that can implement this specification.

We have already shown that a Gaussian lens at Q oriented as (w, l, r) accomplishes a particular non-uniform scaling transformation as well as rotation by π for real images, in accordance with the lens law. We now demonstrate that an arbitrary non-uniform scaling matrix can be implemented by three Gaussian lenses oriented along the x, y and z axes.

Given a 3×3 non-uniform scaling matrix

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$

Consider the lenses l_x with magnification factor m oriented along the x axis, l_y with magnification factor n oriented along the y axis, and l_z with magnification factor p oriented along the z axis. Given s_x, s_y and s_z, we can find solutions to m, n and p by solving

$$\begin{aligned} s_x &= m^2np \\ s_y &= mn^2p \\ s_z &= mnp^2 \end{aligned}$$

Therefore, $s_xs_ys_z = (mnp)^4$, and $mnp = (s_xs_ys_z)^{1/4}$. Now we can solve for m, n and p in terms of the given s_x, s_y and s_z.

Proposition 2 *Lenses and mirrors form a complete set of primitives for achieving object to image transformations that are representable as a product of scaling, rotation, and translation matrices.*

4 An Algorithm for Optical System Design

In this section, we derive a systematic search algorithm for synthesizing optical imaging systems from specifications.

Definition 4 *A composition of primitive optical transforms $T_1, \ldots, T_n$ with associated parameters $\vec{a_1}, \ldots, \vec{a_n}$ implements an optical specification M if*

$$M = T_1(\vec{a_1}) * \ldots * T_n(\vec{a_n})$$

where M is the product of the rotation, scaling and translation transformation matrices derived from the input-output specification.

To better understand how this test for correctness of an implementation can drive a generative algorithm, we show a few steps in the computation of our running example — the design of the imaging system introduced in Section 1. First, we can derive a lower bound on the number and type of components in the design from the specification. Since the determinant of the system specification matrix M is 1, we know from Proposition 1 that we will need a pair of mirrors to implement the rotation. Since a real image is desired, we will need at least one lens. Suppose we consider the design composed of a mirror with normal n_1 at Q_1, a lens at Q with coordinate frame (w, l, r), and focal length F, and another mirror with normal n_2 at position Q_2. The object coordinate frame is u_C and its origin is at u_P. We can write down the following specifications for the components.

$$\begin{aligned} m_1 &= \mathcal{M}_1(n_1, Q_1) \\ l_1 &= \mathcal{L}(Q, (w, l, r), u_P * m_1, F) \\ m_2 &= \mathcal{M}_2(n_2, Q_2) \end{aligned}$$

We write the specification equation from Definition 4

$$m_1 * l_1 * m_2 = ((-1, 0, 0, 0), (0, -1, 0, 0), (0, 0, 1, 0), (1, 1, 0, 1))$$

Each component in the design has its associated constraints. For mirrors, we assert that the vectors n_1 and n_2 are normal vectors. For lenses, the vectors defining the coordinate frame are also normal.

$$\begin{aligned} n_1 * n_1{}^T &= 1 \\ n_2 * n_2{}^T &= 1 \\ w * w^T &= 1 \\ l * l^T &= 1 \\ r * r^T &= 1 \end{aligned}$$

The coordinate frame of the input to the lens must be aligned with the lens coordinate frame. In addition, the lens object and image must lie on the principal axis of the lens.

$$\begin{aligned} m_1 * u_C &= (w, l, r) \\ (Q - (m_1 * u_P)) * l^T &= 0 \\ (Q - (m_1 * u_P)) * r^T &= 0 \\ ((m_1 * l * u_P) - Q) * l^T &= 0 \\ ((m_1 * l * u_P) - Q) * r^T &= 0 \end{aligned}$$

There are 20 parameters to be solved for in order to get a fully instantiated design: 2 parameters each for the two mirror normals and three vectors defining the lens coordinate frame, 3 parameters each for position of the two mirrors and lens, and one parameter for the focal length of the lens. The above equations form an under-constrained system and we need more constraints to determine these parameter values. In our design system, we choose three generic constraint types.

1. We choose a lens coordinate frame and instantiate the principal axis of a lens in the design. In a single lens system, the lens coordinate frame defines the optical axis of the entire system. In optical design, the designer typically specifies the desired principal optical axis of the system. In our running example, we will align the principal axis of the lens to be parallel to the y axis, and set r to be the z axis, since the specification tells us that the object frame is rotated about the z axis.

2. If the object and image are on the same plane, we restrict components in the design to lie on that plane. This is a constraint imposed in practice in the design of optical systems. In our example, the object and image origins lie on the xy plane, so we set the z components of mirror positions, mirror normals and the lens position to be zero.

3. Our models of mirrors and lenses treat them as infinite objects. In practice they are finite and we handle their finite extent by requiring that the components generating inputs and outputs of lenses be aligned along the principal axes of the lens. This ensures that light will reflect off the center of the mirrors and will travel through the center of the lenses.

The first constraint yields

$$(w, l, r) = ((0,1,0),(0,0,1),(1,0,0))$$

and the second constraint yields the following equations,

$$\begin{aligned} n_1 * (0,0,1)^T &= 0 \\ Q_1 * (0,0,1)^T &= 0 \\ n_2 * (0,0,1)^T &= 0 \\ Q_2 * (0,0,1)^T &= 0 \\ Q * (0,0,1)^T &= 0 \end{aligned}$$

Finally, the third constraint yields

$$\begin{aligned} (Q - Q_1) * l^T &= 0 \\ (Q - Q_1) * r^T &= 0 \\ (Q - Q_2) * l^T &= 0 \\ (Q - Q_2) * r^T &= 0 \end{aligned}$$

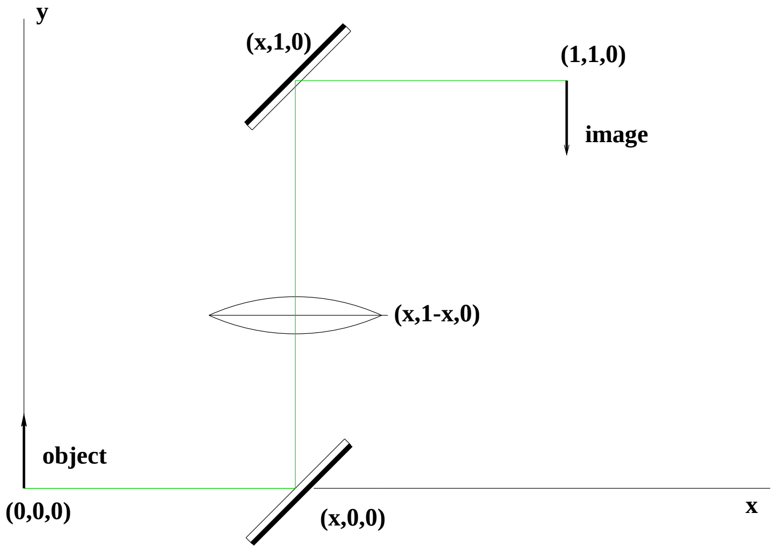

Figure 7: A family of designs for the synthesis problem described in Figure 2. The lens and mirror assembly can move rigidly as a unit anywhere along the x axis in the open interval $(0, 1)$.

These equations are solved by Mathematica's solver (in function `Solve`) to yield the following design.

$$\begin{aligned} n_1 &= (-1/\sqrt{2}, 1/\sqrt{2}, 0) \\ n_2 &= (1/\sqrt{2}, -1/\sqrt{2}, 0) \\ Q_1 &= (x, 0, 0) \\ Q_2 &= (x, 1, 0) \\ Q &= (x, 1-x, 0) \\ F &= 0.5 \end{aligned}$$

This solution encodes a family of designs illustrated in Figure 7 in which the lens and mirrors are all aligned along the y axis, and the whole assembly can move anywhere in the x interval between 0 and 1. If we instantiate x to be 0.5, we obtain the design shown in Figure 2, which is the design of the Xerox 914 series of copiers. The Xerox 914 uses two fixed mirrors and a fixed lens in the above configuration to focus the image of a stationary page laid on the document glass onto a rotating drum.

We have demonstrated the key steps in constructively verifying that a design meets the given specification by solving for a parametric design family.

procedure *Satisfies(Design,Spec,*$(o_P, o_C), (i_P, i_C), l_C$*)*

1. *eqs* $= \{\Pi_{d \in \text{Design}} T_d(\vec{a}_d) = \text{Spec}\}$
2. *add to eqs component constraints for Design:*
    - a. *assert that mirror normals are unit vectors.*
    - b. *assert that lens coordinate frames are composed of unit vectors.*
    - c. *align coordinate frames of object and image of each lens with its coordinate frame.*
    - d. *constrain origin of object and image of each lens to lie on lens principal axis.*
3. *add to eqs constraints that align a lens along the preferred optical axis* l_C.
4. **if** (o_P, o_C) *and* (i_P, i_C) *are in the same plane* **then**
   add to eqs constraints that place all components in Design on that plane.
5. *add to eqs constraints that align components generating inputs and outputs to a lens along the lens' principal axis.*
6. *solutions = Solve(eqs).*
7. **return** *solutions*

end procedure

Table 1: An algorithm for verifying that a design meets the given specification Spec.

These steps include (1) formulating the specification equation, (2) adding the appropriate component level constraints, and, (3) if necessary, adding the three generic refinement constraints, including the one specifying the optical axis of the system, and (4) solving the resulting system of equations. Our representation of the specification of system as well as component behaviors as affine transforms allows us to formulate the synthesis problem as a constraint satisfaction problem and to use symbolic equation solving techniques to derive families of solutions. It is important to see that when we consider a design, e.g., a mirror and a lens followed by a mirror as in this example, we are really considering an infinite number of instantiations of this design family. Further, since we use symbolic equation solving to obtain values of parameters that satisfy the input-output specification and the other applicable constraints, we do not "search" for their values. Therein lies the power of our approach.

The *Solve* function in *Mathematica* is our equation solver. It is a state-of-the-art solver for simultaneous polynomial equations over multiple variables. For simultaneous linear equations, *Solve* uses Gaussian elimination. For polynomial equations in a single variable, *Solve* uses explicit formulas up to degree four, and it attempts to reduce polynomials of degree five and above using factoring and decomposition. *Solve* recognizes and exploits special polynomials. For systems of polynomial equations involving multiple variables, *Solve* constructs a Gröbner basis using an efficient version of the Buchberger algorithm. It is complete, i.e., if there is no solution to a system of equations, *Solve* returns the empty set. *Solve* provides explicit solutions for simultaneous linear equations and a large class of non-linear polynomial equations. When it is unable to find an explicit solution, it reduces the equations to the simplest form.

We now present a systematic search algorithm that uses the above equation solver to verify that a design meets a given specification. The algorithm, shown in Table 2 is an iterative-deepening search algorithm that systematically generates all symbolic sequences of primitive optical components up to the specified depth.

If there are no designs that satisfy the given specification, this algorithm will return the empty set after exploring every possible parametric combination of primitive optical transformations. In the worst case, i.e., when the specification is not realizable given the available primitive components, this algorithm can consider $O(\sum_{i=1}^{Depth} p^i)$ number of choices before declaring failure, where p is the number of primitive optical transformation types, and *Depth* is the maximum size of the designs considered. If mirrors and lenses are the only primitive components considered, then $p = 2$, and if we explore

```
procedure ID_Synthesize(Spec,Depth)
    for d from 0 to Depth by 1 do
        Designs = Synthesize(Spec,d)
        if Designs ≠ [] then return(Designs)
    end for
end procedure

procedure Synthesize(Spec,Depth)
    Solutions = [];
    if Depth = 0 then return Solutions;
    Designs = {{T_i(a_i), 1} | 1 ≤ i ≤ p };
    while (Designs ≠ []) do
            current = Pop[Designs]; des = First[current];
            if Satisfies(des,Spec) then Solutions = Append[{des},Solutions];
            Designs = Append[generate_successors(current,Depth),Designs];
    end while
return(Solutions)
end procedure
```

Table 2: An iterative-deepening generate-and-test algorithm for synthesis of optical systems from specifications of behavior. Spec is the product of the scaling, rotation and translation matrices derived from the object and image configurations. Depth is the bound on the length of designs. The list Designs is initialized to contain p designs each of length 1 containing one of the p available primitive optical components. Generate-successors takes a partial design d of length < Depth and produces p designs of the form $d.T(\vec{a})$ for each primitive optical component $T(\vec{a})$. The computation for checking if a design satisfies a specification is in Table 1.

up to a depth of 4, then $2+4+8+16=30$ parametric combinations of designs up to four components are generated. It is important to realize that while there are 30 symbolic combinations, our design checking procedure implicitly considers an infinity of instantiations of the design family in the solution process.

Since the design testing procedure is complete, we know that if a feasible design exists, the algorithm will find it, i.e., the algorithm is complete. Further, since designs are generated and tested in increasing order of length, when the algorithm returns a set of solutions of a given length, it provides the guarantee that there is no design with fewer components that can achieve the given specification. That is, the algorithm will find the smallest (in the sense of number of components) designs.

Proposition 3 *The synthesis algorithm in Table 2 is complete and optimal in the number of components in the designs. If there is a solution to the specification involving the available primitive optical components, the algorithm will find it. Further, it generates designs with the fewest number of components that achieves the given specification.*

It is not necessary to choose optical axes that are aligned with the x, y or z axes. For instance, if we constrain the lens coordinate frame to be $((-1/\sqrt{2}, 1/\sqrt{2}, 0), (1/\sqrt{2}, 1/\sqrt{2}, 0), (0, 0, 1))$ in our running example, we obtain the design below, illustrated in Figure 3.

$$
\begin{aligned}
n_1 &= (-\frac{\sqrt{2+\sqrt{2}}}{2}, \frac{\sqrt{2-\sqrt{2}}}{2}, 0) \\
n_2 &= (\frac{\sqrt{2+\sqrt{2}}}{2}, -\frac{\sqrt{2-\sqrt{2}}}{2}, 0) \\
Q_1 &= (1, 0, 0) \\
Q_2 &= (0, 1, 0) \\
Q &= (\frac{1}{2}, \frac{1}{2}, 0) \\
F &= \frac{4}{2+\sqrt{2}}
\end{aligned}
$$

Our design algorithm can handle rectilinearly moving objects to a limited extent. For instance, if the object in the above example were moving along the x axis while staying in front of the first mirror, our algorithm re-derives the following design with only one change to the specification. We modify the object's position from $(0,0,0)$, to be $(xx,0,0)$, with the constraint that xx be between 0 and x.

$$
\begin{aligned}
n_1 &= (-1/\sqrt{2}, 1/\sqrt{2}, 0) \\
n_2 &= (1/\sqrt{2}, -1/\sqrt{2}, 0) \\
Q_1 &= (x, 0, 0) \\
Q_2 &= (x, 1, 0) \\
Q &= (x, 1 - x + 0.5xx, 0) \\
F &= 0.5
\end{aligned}
$$

Note that the lens is now moved up on the y axis by half the distance that the object is positioned along the x axis, so that a real image of the same size as the object can be produced. Thus, a lens moving along the positive y axis (within the interval (0,1)) at half the rate of the object's motion along the x axis will preserve the inverted real image of the same size at $(1,1,0)$. Our theory allows us to derive this result with no change to the design algorithm.

5 Experimental Results

We now provide more examples of specifications of copiers in use[4]. We show the designs generated by our system and compare them with the ones that were fielded. We turn from the Xerox 914 to another copier designed and built almost contemporaneously with it. Its specification is:

$$
\begin{array}{ll}
\textit{Input}: & \left\{ \begin{array}{ll} \text{obj pos} & (0,0,0) \\ \text{obj orient} & ((1,0,0),(0,1,0),(0,0,1)) \end{array} \right. \\
\textit{Output}: & \left\{ \begin{array}{ll} \text{img pos} & (4,0,0) \\ \text{img orient} & ((1,0,0),(0,1,0),(0,0,1)) \\ \text{img scale} & ((0,0,0),((1,(1,0,0)),(1,(0,1,0)),(1,(0,0,1)))) \\ \text{real?} & + \end{array} \right.
\end{array}
$$

We choose the optical axis of the system to be aligned with the x axis.

The design family generated by the algorithm in Table 2 is described by the following parameters: it has two fixed mirrors and a lens between them

[4]Subramanian collaborated with the scientists at Xerox Design Research Institute as well as the engineers at the Xerox's Webster production facility to obtain information on the behavioral specifications of copiers.

as follows.

$$
\begin{aligned}
n_1 &= (1/\sqrt{2}, -1/\sqrt{2}, 0) \\
n_2 &= (-1/\sqrt{2}, -1/\sqrt{2}, 0) \\
Q_1 &= (0, y, 0) \\
Q_2 &= (4, y, 0) \\
Q &= (2, y, 0) \\
F &= (y/2 + 1)
\end{aligned}
$$

All three components in the design have the same y coordinate; also note we have a family of designs parameterized by the height along the y axis of all components. This design family is shown in Figure 8. In actual use of this design, the object (page) is rotated about the z axis on a drum centered at $(0, 0, 0)$ and the real image is projected to another drum at $(4, 0, 0)$ rotating about z at the same speed. We also know that there is no design with fewer components that can accomplish this specification.

If we generalize the specification to have the image be positioned anywhere on the x axis by making the image position be $(x, 0, 0)$, the new design derived by our algorithm remains identical to the one above, except that $Q_2 = (x, y, 0)$, $Q = (x/2, y, 0)$, and $F = (y/2 + x/4)$. The lens is positioned halfway between the two mirrors and the focal length is adjusted to accommodate the new path length. This result reflects the generality of the method and points to the ease with which specifications can be modified and designs re-derived.

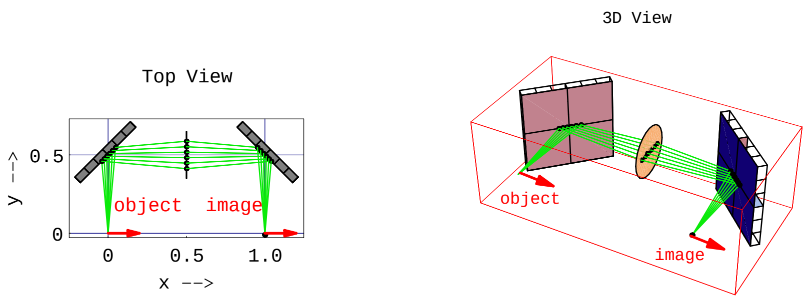

Figure 8: An imaging system designed by our synthesis program. The top view and 3-D arrangement of components are shown. The system transforms an object at (0,0,0) oriented along the x-axis into a real image of the same size at (4,0,0) oriented along the x-axis.

The specification of the more recent Xerox 4000 is shown below.

$$
\begin{array}{ll}
Input: & \left\{ \begin{array}{ll} \text{obj pos} & (0,2,0) \\ \text{obj orient} & ((1,0,0),(0,1,0),(0,0,1)) \end{array} \right. \\
Output: & \left\{ \begin{array}{ll} \text{img pos} & (1,0,0) \\ \text{img orient:} & ((1,0,0),(0,1,0),(0,0,1)) \\ \text{img scale} & ((0,0,0),((1,(1,0,0)),(1,(0,1,0)),(1,(0,0,1)))) \\ \text{real?} & + \end{array} \right.
\end{array}
$$

We specify the optical axis to be parallel to the y axis. Our algorithm derives a design in which the lens axis is oriented along the negative y axis as shown in Figure 9. The light path is folded with two mirrors whose normals are at right angles to one another. We obtain a family of designs, one instantiation of which is shown in Figure 9. The complete design is characterized by

$$
\begin{array}{rcl}
n_1 & = & (\alpha, \sqrt{1-\alpha^2}, 0) \\
n_2 & = & (-\sqrt{1-\alpha^2}, \alpha, 0) \\
Q & = & (0, y, 0) \\
Q_1 & = & (0, y-2+\alpha/\sqrt{1-\alpha^2}, 0) \\
Q_2 & = & (1+\alpha/\sqrt{1-\alpha^2}(yy-y+2), yy, 0) \\
F & = & (4-y)/2
\end{array}
$$

The two mirror normals are at right angles to one another (their dot product is zero). The three free parameters in the design are the angle of inclination of the first mirror, the y position of the lens, and the y position of the second mirror. If we move the object along the y axis and require that the image be projected at $(2,0,0)$, our algorithm derives the requirement that the lens be moved at half the rate of the object.

We tested the performance of our algorithm on a set of imaging system specifications, three of which we have provided complete details on above.[5] The power of our method derives from the fact that our constraint representations allow delayed instantiation of parameters. Each search path encodes an entire family (with an infinity of instantiations) of designs, and symbolic equation solving techniques find constraints on device parameters that make the designs work.

[5] While all examples in this paper were drawn from imaging systems of copiers, the technique is readily applicable to imaging systems of other devices like cameras, periscopes, telescopes and microscopes.

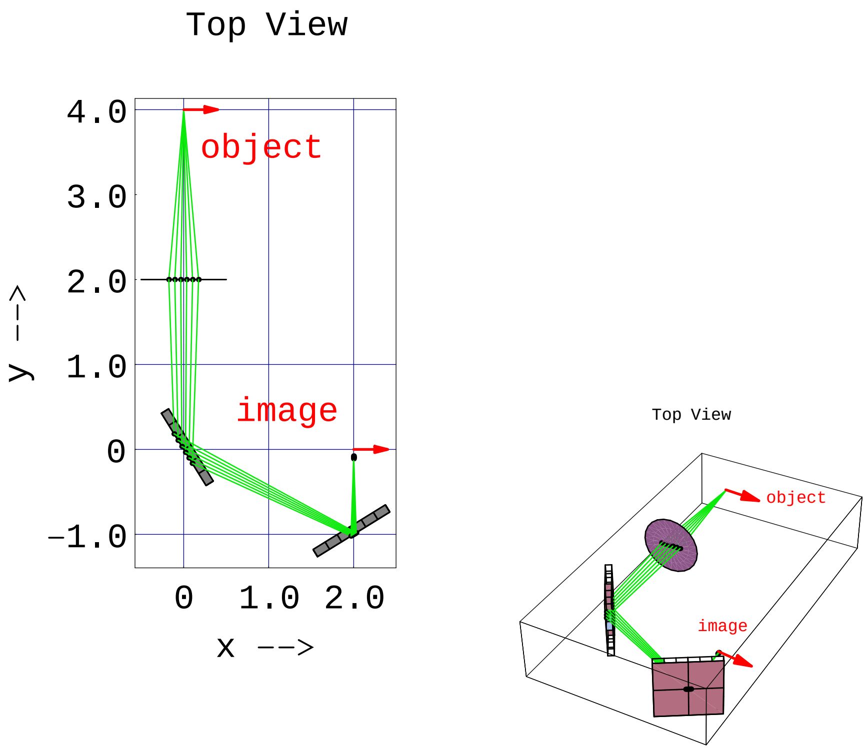

Figure 9: An imaging system designed by our synthesis program. The top view and 3-D arrangement of components are shown. The system transforms an object at (0,4,0) oriented along the x axis into a real image of the same size at (2,0,0) also oriented along the x axis.

We can synthesize stationary assemblies of lenses and mirrors (or equivalently prisms) from specifications of input-output behavior. We can handle moving assemblies to a limited extent – we can move lenses along their axes to project stationary real images when the object is in uniform motion. However, we need to extend our models of mirrors to accommodate their motion. The algorithm in Table 2 still applies with these new component models. Our algorithm performs single-input, single-output synthesis. We can use the extensions in [SW95] to build single input-multi output designs by adding beam splitters to the list of primitive components.

The most important feature of our algorithm is that it is systematic and complete. This means that every feasible topology with a given number of the available primitive components will be generated by the algorithm. By using the iterative-deepening search algorithm in Table 2, we can generate provably minimal designs for given specifications. Our algorithm is very efficient in practice. All of the examples shown in the paper were generated and rendered in milliseconds on commodity hardware.[6] Alternate designs can be explored by changing geometrical preferences, such as the orientation of the optical axis of the system. Constraints are then re-solved by our design algorithm in Table 2. We have demonstrated computationally effective methods to work directly with the laws of optics to generate conceptual layouts of complex optical systems. Our approach works because of three underlying factors: (1) powerful algebraic models of the behavior of optical components, (2) a computational theory of behavior composition, (3) an iterative deepening search algorithm that solves symbolic algebraic equations to allow real-time exploration of conceptual alternatives.

6 Discussion

This paper has presented a solution to the problem of synthesizing conceptual layouts of imaging systems. Here we discuss some of the important properties of our method and the advantages it presents to designers.

First, systematic synthesis techniques for optical imaging systems are not known. Most methods perform synthesis by analysis and require the designer to guess the system's topology — then they solve for the parameters of the components in the guessed topology. Our approach calculates *all* feasible topologies[7] efficiently by symbolically composing behaviors of primitive optical components in a constraint programming environment. We harness advances in techniques for solving algebraic equations as well as available computational power to automate the generation of all conceptual layouts that achieve a given specification. Not only do we compute the topology of the arrangement of components that accomplish a given behavior, we also provide the 3-D locations and orientations of components, making it possible to interface directly to lower level optimization programs that can refine other aspects like shapes and material choice. Our algorithm is easy to use: it only requires specification of the image configuration relative to the object, and some geometric preferences like the orientation of the optical axis

[6] 650MHz Pentium with 256KB RAM running Windows and Mathematica.

[7] Our completeness guarantee is relative to the power of the equation solver, and we are using one of the best solvers known today.

of the system.

Second, we have identified conditions under which our synthesis algorithm produces all feasible designs that can be built from the available primitive components: (1) the system specification must be an affine transformation relating object to image configuration, and (2) the primitive optical components must be representable as affine transformations and form a complete set, and (3) there must be a suficiently powerful equation solver. These conditions are fairly general: under the paraxial approximation in Gaussian optics all components perform affine transformations, and all specifications of optical imaging systems that we have seen can be written as affine transformations on object configurations. The ability to generate all feasible conceptual solutions quickly enhances the efficacy and quality of the final detailed design of imaging systems.

Third, while we have limited our attention in this paper to two basic primitive optical component types (viz., mirrors and lenses) and to single-input single-output systems, the techniques extend to other linear primitive optical component types like beam-splitters, prisms, and fibers. Further, using methods in [SW95], we can extend the algorithm in Table 2 to cover single-input, multiple-output imaging systems by the addition of models for components like beam splitters. We will explore such extensions in future work.

Another important area for extension of our work is the inclusion of obstacles in the problem formulation. Our constraint-based formulation of the synthesis problem yields an unexpected benefit in this context. Since our method is guaranteed to produce symbolic representations of *all* design families built from available primitives that meet the given specifications, we can use information about the position, orientation and extent of obstacles in the space to eliminate infeasible designs in a post-processing step. To accomplish the filtering of infeasible designs, we need to formalize ray paths through the system using generalizations of conventional matrix methods in [GB75]. Then, we can algebraically intersect the geometric constraints that define the obstacle regions with the symbolic ray paths through the system, to constrain the free parameters in the design family. For instance, if there is an obstacle in Figure 2 at (0.2,0.5) oriented along the y axis and with width 0.1, the symbolic constraint solver with the appropriately asserted constraints, will eliminate designs with x in the range (0.15,0.25). This approach of filtering paths after generation is similar to the one adopted in [SC96, SC97] in the context of pipe routing.

6.1 Related work

Our work builds on a strong tradition of research in conceptual design of electro-mechanical systems[FR89, Wil88, WS93, Dix95, Kot89, US89]. We extend it to the optical design arena by modeling optical component behaviors with affine transformations, and by using non-linear algebraic constraint-solving techniques to generate all feasible designs that achieve the specified optical behavior. We draw on three key areas from the literature (1) compositional approaches to configuration design, (2) constraint representations to model component behavior, (3) constraint satisfaction to generate and solve for candidate designs. In this section, we present a brief summary of this related work and place our paper's contribution in context.

Our paper solves a class of configuration design problems, as defined in an insightful essay by Brown [Bro98]. Traditionally, configuration design has been understood as the problem of composing instantiated components from a library of primitives, to implement a given behavioral specification [Dix95]. Brown decomposes configuration design into three subproblems (a) selection of components, (b) association of components into a composite and (c) and evaluation of the composite design. He distinguishes between configuration design problems in which components are fully instantiated [Kot89, WS93, MMD86, SC96], from ones in which they are partially specified [SDS98, SC98]. For the optics problem we examine in this paper, components are parametrically specified, and the design algorithm calculates the unspecified continuous parameters by non-linear constraint solving. The selection of components can be guided by specialized knowledge as in the expert system PRIDE [MMD86], or be based on first-principles as in [SDS98, SW95, SC98]. For conceptual design, the first-principles approach is favored over the expert system approach, which is more appropriate for the detailed design stage. The selection of components for conceptual design is based on the specified behavioral requirements. For instance, Stahovich et. al. [SDS98] use qualitative knowledge of configuration space interactions in a given mechanism sketch to select from a library of abstract mechanisms, while Schmidt and Cagan [SC98] use a high-level energy transformation specification to select from a multi-domain library of components for a drill power train. In this paper, we also use a first-principles approach; however we select components on the basis of the given optical behavior specification, and the properties of the affine transformations implemented by a component class.

There is considerable variation in schemes for representing compositions of selected components; in particular, composition schemes are chosen to reflect underlying regularities in the application design domain. Cagan et. al. [SC96, SC98] who work in pipe routing, truss design and construction of drill power trains, use strings to represent a design. They formulate string rewrite grammars to generate new candidate compositions. Ward and Seering [WS93], who work on instantiating schematic mechanism designs with concrete components drawn from a catalog, use high-level component models and accumulate simple interval constraints on the behavior of components to describe the composite design. Stahovich et. al. [SDS98] as well as Subramanian and Wang [SW95] use similar component models, but their language of qualitative constraints is more expressive. In our work, we use non-linear algebraic equations to describe individual components as well as the interactions between them. We check the consistency of component-level constraints using a non-linear constraint solver. The union of these constraints provides a description of the behavior of the composite design. Our constraints are quantitative instead of qualitative as in [WS93, SDS98, SFH98], and utilize matrix/vector representations of the laws of optics.

Brown formulates configuration design as a dynamic constraint satisfaction problem, in which relationships between chosen components are reasserted dynamically as they are added or removed from consideration in the composite design. We adopt a similar formulation. The inter- and intra-component constraints in the design are recomputed dynamically by our synthesis algorithm. Unlike [SDS98, SFH98], we work with full algebraic constraints and solve the resulting non-linear equations using Gröbner bases.[8] This allows us to construct a complete search algorithm with a powerful constraint solving design verifier at its heart, to generate all feasible designs — a property that cannot be guaranteed by other approaches. Further, our method takes advantage of significant advances in algebraic equation solving and makes it feasible for designers to explore fully specified three-dimensional layouts of stationary as well as moving optical assemblies that meet given specifications.

There are other optical synthesis design systems that can be profitably combined with our first-principles approach for layout synthesis. Expert system techniques such as [Ahm] can be used for adding optical mounts and adjustment mechanisms for layouts produced by our system. The refinement of parameters of lenses in the layout (e.g., material choice, radii, etc.) to meet detailed optical performance specifications can be done by global pa-

[8] [Lak97] is a tutorial account of the use of Gröbner bases for solving nonlinear constraints that arise in the analysis and design of beams and truss structures. It presents Buchberger's algorithm and demonstrates the practicality of the technique in the context of engineering design problems.

rameter optimization methods such as those advocated in [Smi92, Moo99]. The output of our program can serve as input to detail design programs such as OSLO[OSL94] and Winlens[Sch99] which produce designs that can be directly manufactured.

Acknowledgments

I thank the National Science Foundation for funding this work with grant IRI-9796046, and my program director Dr. Ephraim Glinert for his continued support and encouragement.

References

[Ahm] Anees Ahmad. Knowledge-based expert system for optomechanical design. Technical Report 3130-15, Applied Optics Lab, Univ. of Arizona.

[BP91] P.P. Banerjee and T-C. Poon. *Principles of applied optics.* Irwin Press, 1991.

[Bro98] D. C. Brown. Defining configuring. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 12:301–305, 1998.

[Dix95] J. R. Dixon. Knowledge-based systems for design. *Transctions of the ASME: Journal of Mechanical Design*, 117:11–16, 1995.

[FR89] S. Finger and J. Rinderle. A synthesis strategy for mechanical devices. *Research in Engineering Design*, 1(2):87–104, 1989.

[GB75] A. Gerrard and J. M. Burch. *Introduction to matrix methods in optics.* John Wiley and Sons, 1975.

[Gol91] R. N. Goldman. Recovering the data from the transformation matrix. In James Arvo, editor, *Graphics Gems II*, pages 324–331. Academic Press, 1991.

[Gol94] R. N. Goldman. The algebra of points and vectors. Technical report, Siggraph tutorial, 1994.

[Kot89] S. Kota. Qualitative motion synthesis: toward automating mechanical systems configuration. In *NSF Design and Manufacturing Systems Conference*, pages 77–91, 1989.

[Lak97] Sivand Lakmazaheri. Constraint-based reasoning via grobner bases. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 11:5–15, 1997.

[Lev68] Leo Levi. *Applied Optics: Vol 1: a guide to optical system design.* John Wiley and Sons, 1968.

[MMD86] S. Mittal, M. Marjoria, and C. L. Dym. Pride: an expert system for the design of paper handling systems. *IEEE Computer*, 19(7):102–114, 1986.

[Moo99] Kenneth E. Moore. Algorithms for global optimization of optical systems based on genetic competition. In Richard C. Juergens, editor, *Optical design and analysis software, Proceedings of the SPIE*, pages 40–47. SPIE Press, 1999.

[Nus98] Allen Nussbaum. *Optical System Design.* Prentice Hall, 1998.

[O'S85] Donald O'Shea. *Elements of modern optical design.* John Wiley and Sons, 1985.

[OSL94] OSLO. User's manual for oslo light, v5.2. Technical report, Sinclair Optics, 1994.

[Pen96] D. Penkethman. Optomechanical tolerancing by means of system modeling with affine transformations. In J.J.M. Braat, editor, *SPIE Proceedings on Design and Engineering of Optical Systems*, pages 116–133, 1996.

[Pet82] H. Petroski. *To engineer is human: the role of failure in engineering design.* St. Martin's Press, New York, 1982.

[SC96] S. Szykman and J. Cagan. Synthesis of optimal non-orthogonal routes. *Transctions of the ASME: Journal of Mechanical Design*, 118:419–424, 1996.

[SC97] S. Szykman and J. Cagan. Constrained 3d component layout using simulated annealing. *Transctions of the ASME: Journal of Mechanical Design*, 119:28–35, 1997.

[SC98] L. Schmidt and J. Cagan. Optimal configuration design: an integrated approach using grammars. *Transctions of the ASME: Journal of Mechanical Design*, 120:2–9, 1998.

[Sch99] R. G. Schuhmann. Low cost analysis software for optical design. In Richard C. Juergens, editor, *Optical design and analysis software, Proceedings of the SPIE*, pages 48–55. SPIE Press, 1999.

[SDS98] T. F. Stahovich, R. Davis, and H. Shrobe. Sketchit: generating multiple new designs from a sketch. *Artificial Intelligence*, 104(1-2):211–264, 1998.

[SFH98] M. Stumptner, G. E. Friedrich, and A. Hasebock. Generative constraint-based configuration of large technical systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 12:307–320, 1998.

[Smi92] Warren J. Smith. *Modern Lens Design: A resource manual.* McGraw Hill, 1992.

[SW95] D. Subramanian and Edward Wang. Kinematic synthesis using configuration spaces. *Research in Engineering Design*, 7:193–213, 1995.

[US89] K. Ulrich and W. P. Seering. Synthesis of schematic descriptions of mechanical design. *Research in Engineering Design*, 1:3–18, 1989.

[Wil88] B. Williams. *Principed design based on topologies of interaction.* PhD thesis, Department of EECS, MIT, 1988.

[WS93] A. C. Ward and W. P. Seering. The performance of a mechanical design compiler. *Transctions of the ASME: Journal of Mechanical Design*, 115:341–345, 1993.