

Generalized As-Similar-As-Possible Warping with Applications in Digital Photography

Renjie Chen¹ and Craig Gotsman²

¹ Max Planck Institute for Informatics, Saarbrücken, Germany

² Jacobs Technion-Cornell Institute, Cornell Tech, New York



Figure 1: (left) Vintage photo taken in 1911 of Broadway at W. 72nd St. in New York City, courtesy of mycvintageimages.com. (center) Contemporary photo taken of the same scene in 2014. Corresponding line segment features are marked in red on both photos. Note that the ornate building in the center background appears larger in the vintage photo, yet the 72 St. subway station in the center foreground is smaller. (right) The contemporary photo is warped using an instance of our generalized ASAP method (called AAAP) to align it with the vintage photo and then combined with the vintage photo to illustrate the quality of the alignment.

Abstract

Discrete conformal mappings of planar triangle meshes, also known as the As-Similar-As-Possible (ASAP) mapping, involve the minimization of a quadratic energy function, thus are very easy to generate and are popular in image warping scenarios. We generalize this classical mapping to the case of quad meshes, taking into account the mapping of the interior of the quad, and analyze in detail the most common case - the unit grid mesh. We show that the generalization, when combined with barycentric coordinate mappings between the source and target polygons, spawns an entire family of new mappings governed by quadratic energy functions, which allow to control quite precisely various effects of the mapping. This approach is quite general and applies also to arbitrary planar polygon meshes.

As an application of generalized ASAP mappings of the unit grid mesh, we demonstrate how they can be used to warp digital photographs to achieve a variety of effects. One such effect is modifying the perspective of the camera that took a given photograph (without moving the camera). A related, but more challenging, effect is re-photography – warping a contemporary photograph in order to reproduce the camera view present in a vintage photograph of the same scene - taken many years before with a different camera from a different viewpoint. We apply the generalized ASAP mapping to these images, discretized to a unit grid. Using a quad mesh (as opposed to a triangle mesh) permits biasing towards affine maps of the unit squares. This allows the introduction of an As-Affine-As-Possible (AAAP) mapping for a good approximation of the homographies present in these warps, achieving quite accurate results. We demonstrate the advantages of the AAAP mapping on a variety of synthetic and real-world examples.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications

1. Introduction

Spatial warps controlled by user-provided constraints are an important tool in digital geometry, used in the 2D setting to modify images, and in the 3D setting to modify, edit and animate 3D models. The warps are functions $f: R^d \rightarrow R^d$, and are either defined on a continuous domain $D \subseteq R^d$, or on a discretization of D . For example, in 2D, the discretization could be a planar triangulation of D , e.g. a so-called *source triangulation* T , and the objective would be to determine the images of the vertices of T – the so-called *target triangulation*. The interior of each source triangle is then mapped affinely to its target counterpart, defining the image of every point in D .

A popular method to warp a planar triangle mesh is the so-called As-Similar-As-Possible (ASAP) mapping, first coined by Liu et al. [LZX*08] in a 3D to 2D parameterization context, but already implicit in the image morphing and warping work of Alexa et al. [ACL00] and Igarashi et al. [IMH05], and in the discrete conformal parameterization work of Levy and Maillot [LM02] and Desbrun et al. [DMA02]. In a nutshell, this method tries to map each triangle of T to a target which is, as much as possible, a similar copy of itself, subject to the user-supplied constraints. It is a discrete approximation to the well-known conformal map (preserving angles) from complex analysis, and may be easily computed by solving a linear system in the target vertex positions of T , derived from the minimization of a quadratic energy function (called the *ASAP energy* or *conformal energy*). This is quite an effective mapping and has been used since in many common image warping applications.

The ASAP mapping of a planar triangle mesh (and its more sophisticated variant – the As-Rigid-As-Possible (ARAP) mapping [ACL00, IMH05]) – is well understood in the triangle mesh case. Since we typically use a so-called *barycentric* mapping between a source polygon and its target counterpart, in the triangle mesh setting, as soon as the target triangulation vertices have been specified, the mapping of the source domain D is well-defined and unique – the piecewise affine mapping between each source triangle and its target counterpart. However, in the (actually more common) case where the source domain D is discretized by a general polygonal mesh, specifying just the positions of the target mesh vertices is not sufficient to define the mapping of D , since each source polygon can be mapped to its target counterpart by many possible mappings. It turns out that each such barycentric mapping induces a different ASAP quadratic energy function, whose minimization affects the positions of the target mesh vertices. For the case of a quad mesh, the quadratic energy function is governed by a single parameter, trading off various biases in the mapping, and for a mesh consisting of general polygons, there are additional degrees of freedom. In this paper we concentrate on quad tilings, in particular the common unit square grid, for which we provide a complete analysis.

The degree of freedom present in the ASAP warp of the unit square grid allows us to tune the resulting warps to align them better to the nature of the warps needed in specific digital photography applications, and even bias them *away* from the traditional conformal-type mapping. The two related applications that we explore in this paper involve synthetically changing the camera used to take a digital photograph. In the first application, we would like to change the perspective of the camera (e.g. from wide angle to narrow angle) without moving it. It is well known that this change transforms the images of planar regions in the scene through a homography [HZ04], which is a variant of a projective transformation. In many

situations, this may be approximated well by a linear affine transformation, which is much simpler than the non-linear projective transformation, so it is particularly advantageous to bias the discrete map to have a local affine character in these settings. We call this variant of the ASAP mapping the As-Affine-As-Possible (AAAP) mapping. An example of this application appears in Fig. 2. The second application, called re-photography, is more challenging. It involves warping a contemporary photograph in order to reproduce the camera view present in a vintage photograph of the same scene taken many years before from a similar viewpoint. In contrast to the first application, here not only is the camera perspective different between images, but usually also the precise camera position and orientation. The fact that the scene geometry may have changed during the years between the two shots makes for a particularly challenging situation. We show that user guidance, in the form of corresponding line segments between the two photographs, is sufficient to guide the AAAP warp to achieve quite a pleasing result, with very minor distortions and artifacts. Fig. 1 shows an example.

2. Related Work

2.1. Planar Warps

Planar image warping has been studied extensively over the last decade and we mention here just the most noteworthy techniques. Early methods guided by point-to-point correspondences, such as those based on Radial Basis Functions (RBF) [Boo89] express the warp as a linear combination of quite complicated “basis functions” in each dimension. The result is smooth, but it is notoriously difficult to predict how it will behave away from the (correspondence) data. This method does not seem to conform to any underlying geometric model or assumptions. More recent popular techniques are *cage-based* (essentially mapping the interior of a polygon) deformation methods employing barycentric coordinates (e.g. [Flo03, JMD*07]), their more sophisticated variants (e.g. JBPS11]), and the Moving Least Squares (MLS) techniques [SMW06]. The latter apply spatially-varying affine or similarity transformations, computing them as locally “optimal” maps in a weighted least-squares sense based on the constraints, where the weights are inversely proportional to the distance of the point from each constraint.

2.2. Conformal Maps

In many applications, a conformal mapping is considered ideal, since it preserves the local geometry of the source plane, essentially mapping each differential element to a similar (translated, scaled and rotated) version of itself. While pure conformal mappings of planar regions have been studied extensively in the branch of mathematics known as complex analysis and are well understood, in practice various application-specific constraints on the mapping prevent it from being conformal. Thus, the objective is typically to approximate a conformal map as much as possible, while satisfying the constraints, minimizing some measure of non-conformality. There are many ways to measure the non-conformality of a planar mapping, all involving the Jacobian of the mapping, and the so-called *Cauchy-Green tensor*. We refer the interested reader to [FH05, MRT12, KWS*13] for a complete treatment and description of possible *conformal energies* – measuring how distant the mapping is from a conformal map. In our work we use the simplest and most classical definition of conformal energy which, conveniently, is a quadratic function of the problem parameters.

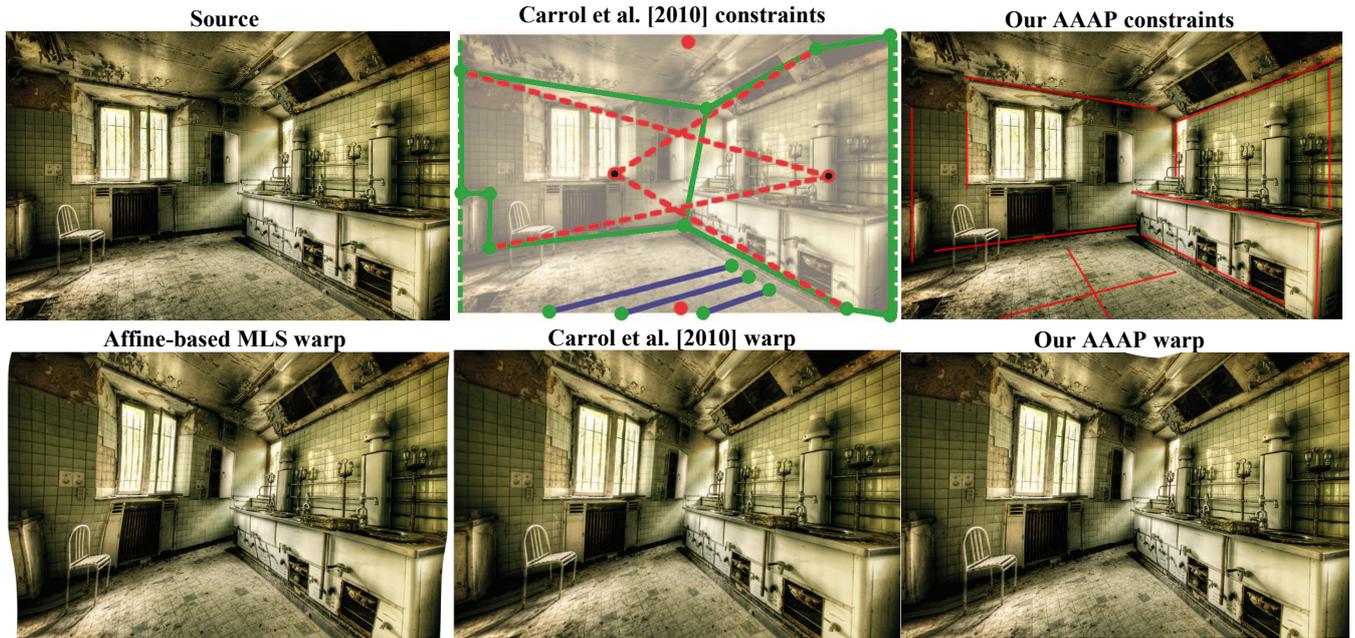


Figure 2: Comparison between warps attempting to change the perspective of the camera of a photograph. **(Top left)** The input (source) image. **(Top middle)** The constraints entered by the user in the system of Carrol et al [CAA10]. Green lines bound polygons which are transformed as planar homographies. Dashed red lines meet at new vanishing points. Blue lines are line constraints. **(Top right)** The constraints entered by the user in our generalized ASAP deformation system. All red lines are linear constraints, which is all that is required. **(Bottom left)** Resulting warp generated by affine-based MLS with the linear constraints used in our AAAP warp. Note how many of the unconstrained vertical lines are no longer vertical, sometimes not even straight. **(Bottom middle)** Warp generated by mapping of Carrol et al. [CAA10], requiring a few seconds to render. **(Bottom right)** Our AAAP warp, generated in real-time.

2.3. Discrete Planar Warps

Planar warps may also be discretized, the most popular method being to compute the map of a triangulation of the domain, itself a triangulation, thus defining a piecewise-affine map between the interiors of each source triangle and its image. To approximate a conformal map, the As-Similar-As-Possible (ASAP) map [LZX*08] (also called the Least-Squares Conformal Map (LSCM) by Levy and Maillot [LM02] and the Discrete Natural Conformal Parameterization (DNCP) by Desbrun et al. [DMA02]), tries to map each source triangle to a target triangle which is a similar (i.e. a scaled, rotated and translated) copy of itself, subject to positional constraints on some of the triangulation vertices. This has been shown equivalent to minimizing a quadratic energy, thus reduces to a linear system involving the positions of the triangulation vertices. The more elaborate As-Rigid-As-Possible (ARAP) map [ACL00, IMH05] has the similar objective of mapping each triangle to a rigid (i.e. a rotated and translated) copy of itself. In contrast to ASAP, this is a non-linear problem, the most popular solution being the so-called local-global iterative algorithm [LZX*08] and its variants [KFG09].

Despite the simplicity of working with a triangle mesh, many image processing algorithms prefer to work with a mesh of quadrilateral elements, in particular the special case of the square grid, which is what we deal with here. The family of warps that we introduce on these meshes is closely related to the simple ASAP map of a triangle mesh. We show that this family of maps is governed by a quadratic energy function with one degree of freedom. This degree of freedom may be used to bias the mapping in different ways. So although zero energy will be achieved only by a mapping for which

each source square is mapped to an exact target square, deviations from this ideal mapping can be penalized in different ways, all falling within the framework of a quadratic energy function. This provides a particularly versatile method of designing planar warps.

2.4. Mapping a single quad

The square grid is the preferred mesh for many image processing applications, due to its simplicity, symmetry, uniformity and axis-aligned elements. It has become most popular in *image resizing* applications, where the objective is to modify the aspect ratio of an image (typically for different video formats), such that the image content undergoes minimal distortion. Some of these image resizing applications require that the square elements stay axis-aligned, thus do not allow rotation [Wang et al. 2008, Panozzo et al. 2012], while some do, preferring a conformal-type behavior [ZCHM09].

The easiest way to treat a quad in a discrete conformal mapping scenario is as two separate triangles formed by a diagonal. This reduces the quad case to the triangular case. While convenient, this is probably the worst possible method, since the resulting piecewise-affine mapping will inevitably introduce an unnecessary discontinuity along the diagonal.

Some authors [GSC06, ZCHM09] treat the quad as just four points in the plane (the four vertices of the quad), and define conformal energy to be the *Procrustes distance* under a similarity transformation of the four source points to their images. To be more concrete, assume the four vertices of the *source* quad, expressed as homogeneous complex numbers, are arranged in a 4×2 matrix \mathbf{p}

and are mapped to the four vertices of the *target* quad \mathbf{q} , represented as a 4×1 vector, i.e.

$$\mathbf{p} = \begin{pmatrix} p_1 & 1 \\ p_2 & 1 \\ p_3 & 1 \\ p_4 & 1 \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix},$$

The similarity-based Procrustes distance between \mathbf{p} and \mathbf{q} is the quadratic form

$$D(\mathbf{p}, \mathbf{q}) = \mathbf{q}^* \mathbf{C} \mathbf{q}, \quad \mathbf{C} = \mathbf{I} - \mathbf{p} \mathbf{p}^+ \quad (1)$$

where \mathbf{q}^* is the conjugate transpose of \mathbf{q} and \mathbf{p}^+ is the 2×4 pseudo-inverse of \mathbf{p} . Although we have written this for a quad, the same formula applies also to a polygon \mathbf{p} with any number of sides n , (by essentially treating \mathbf{p} and \mathbf{q} as sets of n points, and \mathbf{C} is a $n \times n$ matrix).

As we will see later, this is a rather simplistic way to treat a quad, as it does not take into account many geometric aspects of the quad, such as its size or area. It also does not take into account the interior of the quad in any way, in particular how the interior of \mathbf{p} is mapped to the interior of \mathbf{q} , e.g. using the popular bilinear mapping or something else. The more general formula which we develop later will take all of these into account. Moreover, it will also allow us to modify the energy of a square element so that, instead of it favoring a conformal (square) image of the square, it will actually allow shear in the mapping, preferring a pure affine map of the square to a parallelogram. This will allow us to achieve piecewise-affine maps of the square elements, which provide an excellent approximation to important classes of popular maps. We call this mapping the As-Affine-As-Possible (AAAP) mapping.

2.5. Application I: Perspective Manipulation

Frequently, photographs are taken with the camera that happens to be available, which does not always do justice to the scene or capture it as desired by the photographer. A change of perspective is sometimes required. This problem was considered by Carrol et al. [CAA10], who described a system to warp a photograph, driven by user-supplied directives such as the positioning of line constraints, planar homographies and vanishing points, to achieve this effect. A variant of this method has been implemented as the ‘‘Perspective Warp’’ feature in Adobe Photoshop (see <https://helpx.adobe.com/photoshop/using/perspective-warp.html>) While generating quite good results, computing this perspective warp involves intricate user interaction and the solution of a large non-linear and non-convex optimization problem, which can be complicated and time-consuming. Since perspective warps may be approximated well by affine transformations in many cases, we suggest an alternative based on our quadratic AAAP energy on a square grid with just linear constraints, which achieves very similar results with a simpler user interface and a very fast linear solver, resulting in a real-time interactive experience. See Fig. 2 for an example. A related application which works on a collection of images taken from different viewpoints and transfers features between the images was considered by Yucer et al. [YJHS12].

2.6. Application II: Re-Photography

Another digital photography application that AAAP may be applied to is re-photography, which can be loosely defined as follows: given a vintage photograph of an urban scene, typically taken in the distant past, try and shoot a contemporary photograph of the same scene which is as close as possible to the vintage photograph, in terms of the camera position, orientation and other parameters. Fig.

1(left) shows an example of a vintage photograph taken in New York City in 1911, and Fig. 1(center) an attempt to reproduce the same photograph in 2014. This is a difficult problem in general, not only because much of the scene content has changed over the years, but also because identifying the exact location, orientation and perspective of the original camera is non-trivial, and the result may be quite sensitive to this. On top of this, differences between the camera optics (e.g. field of view, focal length, aberrations), which are typically out of the photographer’s control, make it virtually impossible to reproduce the original photograph. Only trial and error, access to quality cameras, and a lot of patience can result in somewhat reasonable results. Thus re-photography remains an art painstakingly practiced by professional photographers over the years. An impressive example is the work of Douglas Levere, who, circa 2000, re-photographed scenes of New York City taken by Berenice Abbott circa 1938. Sample photos by Levere, side-by-side with those of Abbott, may be seen in Levere’s book [Lev04], or online at www.newyorkchanging.com. Another re-photography project [Que05] depicts photos of Paris a hundred years apart.

A notable work in the field of interactive digital re-photography is that of Bae et al. [BAD10], who describe a system consisting of a digital camera connected to a laptop which attempts to instruct the user (in real-time) how to reposition the camera in order to best match to a given vintage photograph. With today’s technology, this entire system may be implemented on a smartphone, and, indeed, a number of smartphone apps support simple re-photography, the most popular being Re-photo, DejavuCam, RePrism and Timera. Alas, despite these handy digital re-photography aids, there are still so many issues beyond the positioning of the camera that post-processing of the result is usually still necessary.

Since physical re-photography in general is quite difficult, and for most applications it is important to have very accurate results, it would be very helpful to be able to digitally ‘‘fix’’ one of the images (typically the contemporary one) so that it coincides better with the other, namely warp the image based on corresponding features in both images. In theory this could be done automatically by a digital computer which identifies the correspondences and then warps the image accordingly. In practice it is not so easy, as automatic correspondence-finding algorithms work well with images which are very similar (e.g. stereo pairs, successive frames of video), but not when the images have much less in common. Lee et al. [LLC11] address the re-photography problem using *multiple* images collected from different viewpoints close to that of the vintage photo. They then resort to stereo techniques to build a 3D point cloud, of which some are *manually* matched to the vintage photo, allowing to compute the position of the vintage camera. The software then combines all the input images through a combination of *forward warping* and *backward warping* using the vintage camera position to generate the final result.

We propose a pure 2D image warping technique for re-photography. We assume that the feature correspondence between images is done manually, with a user visually matching between the two. As others have done before [SMW06, CAA10], we use line segments to guide the warp, since these are the predominant features in urban photographs (e.g. building edges, corners and silhouettes). Thus the main remaining problem, the one we address in this paper, is the image warp based on this given correspondence information.

A very popular photographic application related to re-photography is that of stitching panoramic images, each taken from a slightly different camera viewpoint. A good result requires a warp of the

images in a manner that compensates for these differences. This is easier than re-photography since the scene is identical in all images, and typically the camera moves along a continuous trajectory. Motivated by the fact that planar homographies may be approximated well by affine transformations, Lin et al. [LLM*11] proposed doing this with an approach similar to affine-based MLS. More recently, Zaragoza et al. [ZCBS13] developed a perspective-based variant of MLS.

3. The Quadratic Conformal Energy

In this section we develop the theory for constructing quadratic conformal energy functions for discrete 2D mappings. While the theory is applicable to any planar polygonal mesh, we are mainly interested in a unit square grid of finite resolution. The map is such that each square in the source domain is mapped to a quad in the target domain, satisfying the user-supplied constraints and minimizing some energy function, which is quadratic in the positions of the target vertices. We start by examining the simplest case of the mapping of just a single square.

3.1. A single square

It is convenient to work with complex numbers in lieu of 2D vectors. Consider the following four (column) vectors in \mathbb{C}^4 :

$$\begin{aligned} \mathbf{o} &= (1, 1, 1, 1)^t & \mathbf{s} &= (1, i - 1, -i)^t \\ \mathbf{a} &= (1, -1, 1, -1)^t & \mathbf{b} &= (1, -i - 1, i)^t \end{aligned}$$

which are the orthogonal Fourier basis of \mathbb{C}^4 , based on the 4th roots of unity. Note that \mathbf{o} is the all-ones vector and that \mathbf{s} describes a square S with vertices in counter-clockwise order. Note also that any complex vector of the form $(x, y, z, x + z - y)^t$ is orthogonal to \mathbf{a} , and any complex vector of the form $(x, y, z, -ix + iz + y)^t$ is orthogonal to \mathbf{b} (remember that two complex vectors \mathbf{v}_1 and \mathbf{v}_2 are orthogonal iff $\mathbf{v}_1^* \mathbf{v}_2 = 0$, where \mathbf{v}_1^* is the conjugate transpose of \mathbf{v}_1). When interpreted as the vertices of quads in counter-clockwise order, these orthogonal vectors have interesting shapes. The first represents the set of parallelograms – quads whose two pairs of opposite edges have equal lengths, and the second represents the set of crossgrams - quads whose two diagonals are perpendicular and have equal lengths. The intersection of these two sets is just the set of squares.

Denote by \mathbf{A} and \mathbf{B} the following 4×4 matrices: $\mathbf{A} = \mathbf{a}\mathbf{a}^*$, $\mathbf{B} = \mathbf{b}\mathbf{b}^*$, and define $\mathbf{S}_{\alpha,\beta} = \alpha\mathbf{A} + \beta\mathbf{B}$ for any real positive α and β . Note that both \mathbf{A} and \mathbf{B} are rank-1, Hermitian and circulant (see Fig. 3), thus $\mathbf{S}_{\alpha,\beta}$ is rank-2, Hermitian and circulant, and its nullspace is spanned by \mathbf{o} and \mathbf{s} . Now consider this quadratic energy function on \mathbb{C}^4 :

$$E_{\alpha,\beta}(\mathbf{q}) = \mathbf{q}^* \mathbf{S}_{\alpha,\beta} \mathbf{q} \quad (2)$$

If we interpret $\mathbf{q} = (q_1, q_2, q_3, q_4)^t$ as the (counter-clockwise) vertices of an arbitrary quad in column vector form, then for any $\alpha \neq 0$, $\beta \neq 0$, $E_{\alpha,0}(\mathbf{q}) = 0$ iff \mathbf{q} is a parallelogram, $E_{0,\beta}(\mathbf{q}) = 0$ iff \mathbf{q} is a crossgram and $E_{\alpha,\beta}(\mathbf{q}) = 0$ iff \mathbf{q} is a square. This implies that $E_{\alpha,0}(\mathbf{q})$ can be used as a quadratic energy function to force the image \mathbf{q} of \mathbf{s} to be a parallelogram, i.e. an affine mapping, and $E_{\alpha,\beta}(\mathbf{q})$ may be used to force the image to be a square, i.e. a similarity, which is a conformal map. The different combinations of α , β balance the preference between the target being a parallelogram, a crossgram, or a square.

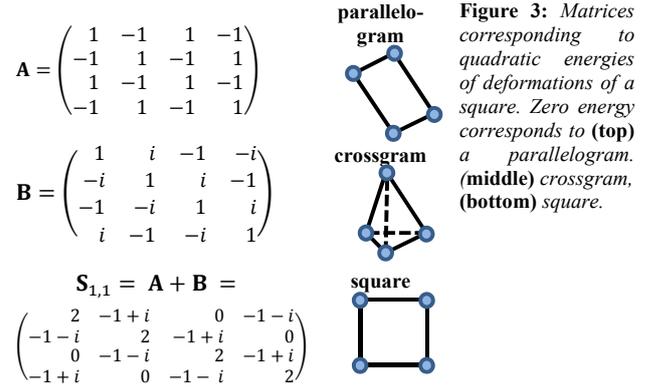


Figure 3: Matrices corresponding to quadratic energies of deformations of a square. Zero energy corresponds to (top) a parallelogram, (middle) crossgram, (bottom) square.

It turns out that for certain values of α and β , $E_{\alpha,\beta}$ measures the conformal energy of a mapping of the interior of a square S to the interior of an arbitrary planar quad Q . The precise values α and β used in $E_{\alpha,\beta}$ will depend on the particular mapping of the interior of S . Assume that the mapping f of the interior of S to the interior of Q is by a barycentric coordinate scheme, namely, if $z = \sum_{i=1,\dots,4} B_i(z) s_i$, then $f(z) = \sum_{i=1,\dots,4} B_i(z) q_i$, with the (standard) assumption that f maps the edges of S linearly to the edges of Q . By definition, the conformal energy of f is:

$$E_c(f) = 2 \iint_S \left| \frac{\partial f}{\partial \bar{z}} \right|^2 dz \wedge d\bar{z}$$

Let us consider two special barycentric maps. The first map is reduction to triangles, namely, S is naively split into two triangles, and each is mapped affinely to the corresponding triangle of Q . In this map, one of the four barycentric coordinates is always zero. The second map is the bilinear map between S and Q .

As we show in the Appendix, the conformal energy of *any* barycentric mapping of the unit square S to a quad Q can be expressed as a quadratic energy of the form (2). The two-triangle mapping corresponds to the case $\alpha = \beta = 1/4$, and the bilinear mapping (which is also a harmonic mapping), corresponds to the case $\alpha = 1/12$, $\beta = 1/4$. Note that the 4×4 matrix \mathbf{C} of (1) is, in this case, equivalent to $\alpha = \beta = 1/4$, which, up to a constant, assumes the naive two-triangle mapping. Also note that in this case, the matrix contains zeros at the four entries corresponding to the two diagonals of the square S , in contrast with the harmonic (or other) case, where the matrix is dense (this is consistent with the findings of Alexa and Wardetsky [AW11] for the Laplacian of polygonal meshes). Moreover, note that these values of α and β imply that the conformal energy of the bilinear (harmonic) map of S to *any* Q is *always* less than that of the two-triangle map, which is also consistent with the fact that the harmonic mapping minimizes the Dirichlet energy (which is equal to the conformal energy plus the constant target quad area).

3.2. The unit grid

Now assume that the domain is discretized to a square grid G of n vertices in the plane, whose coordinate vectors are the $n \times 1$ column vector \mathbf{g} . For given values of α and β , the procedure of the previous paragraph may be applied to each unit square S of G , yielding a 4×4 matrix $\mathbf{S}_{\alpha,\beta}$ per square. These individual 4×4 matrices may then be accumulated (by summing) into a large sparse $n \times n$ matrix $\mathbf{L}_{\alpha,\beta}$, where $\mathbf{S}_{\alpha,\beta}$ affects the 16 entries of $\mathbf{L}_{\alpha,\beta}$ corresponding to the 16 pairs of vertices of S . In other words, assume

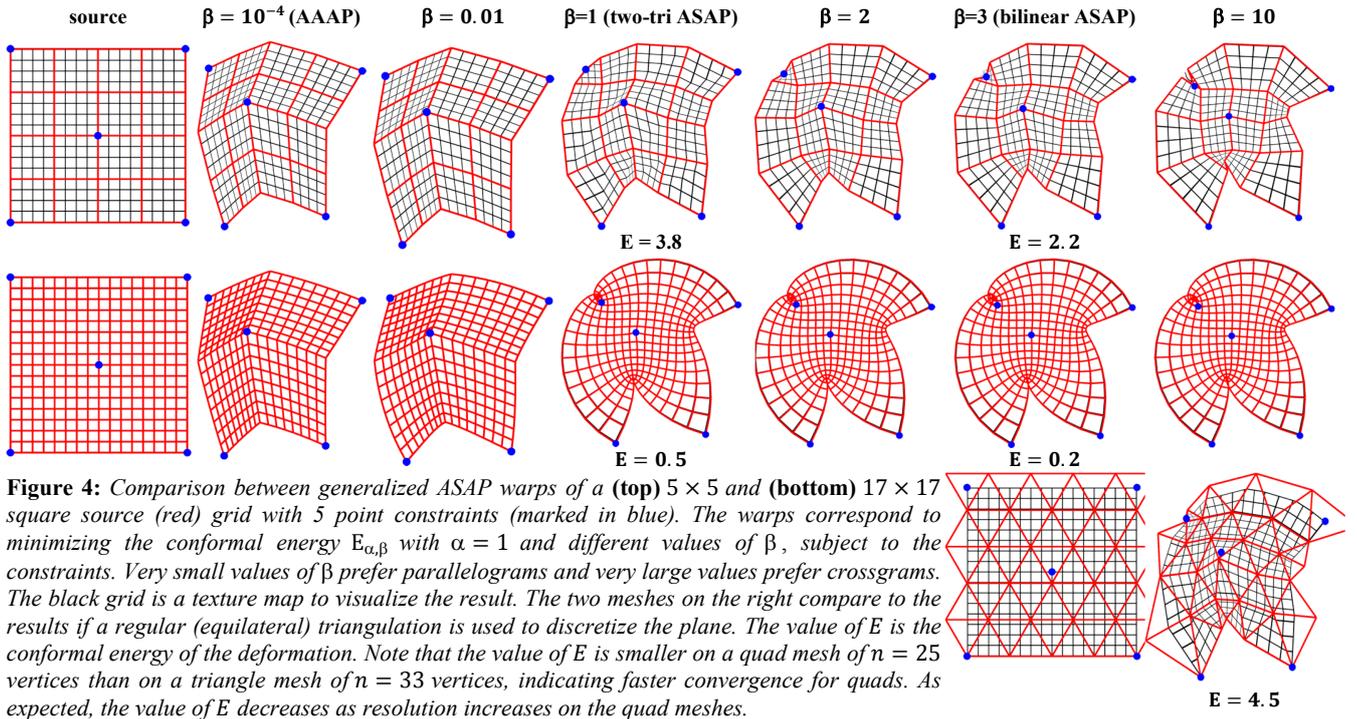


Figure 4: Comparison between generalized ASAP warps of a (top) 5×5 and (bottom) 17×17 square source (red) grid with 5 point constraints (marked in blue). The warps correspond to minimizing the conformal energy $E_{\alpha,\beta}$ with $\alpha = 1$ and different values of β , subject to the constraints. Very small values of β prefer parallelograms and very large values prefer crossgrams. The black grid is a texture map to visualize the result. The two meshes on the right compare to the results if a regular (equilateral) triangulation is used to discretize the plane. The value of E is the conformal energy of the deformation. Note that the value of E is smaller on a quad mesh of $n = 25$ vertices than on a triangle mesh of $n = 33$ vertices, indicating faster convergence for quads. As expected, the value of E decreases as resolution increases on the quad meshes.

that the 4 vertices of a square S have indices i_1, i_2, i_3 , and i_4 in G . Then

$$\mathbf{L}_{\alpha,\beta}(i_k, i_m) \leftarrow \mathbf{L}_{\alpha,\beta}(i_k, i_m) + \mathbf{S}_{\alpha,\beta}(k, m) \quad 1 \leq k, m \leq 4$$

The energy of the deformation of G to another quad grid H having geometry \mathbf{h} will then be

$$E_{\alpha,\beta}(\mathbf{h}) = \mathbf{h}^* \mathbf{L}_{\alpha,\beta} \mathbf{h} \quad (3)$$

Since $E_{\alpha,\beta}(\mathbf{h})$ measures how close the shapes of the quads in H are to being squares, we call $E_{\alpha,\beta}$ the As-Similar-As-Possible (ASAP) energy. Recall that if $\beta = 0$, $\mathbf{S}_{\alpha,\beta}$ will be rank 1, and the resulting $\mathbf{L}_{\alpha,0}$ will measure how close the shapes of the quads of H are to being parallelograms. We call the special case of $E_{\alpha,0}$ the As-Affine-As-Possible (AAAP) energy. In general, taking β very small relative to α biases the energy to prefer a deformation in which the images of the squares of G are close to parallelograms. It is important to emphasize that the AAAP deformation **cannot** be achieved at all using a triangle grid, simply because it does not possess the required extra degrees of freedom.

$\mathbf{L}_{\alpha,\beta}$ has some interesting properties: $\mathbf{L}_{\alpha,\beta}$ is Hermitian, and all entries but those corresponding to boundary edges of G are real. $\mathbf{L}_{\alpha,\beta}$ is sparse, namely, all entries corresponding to pairs of grid vertices which are not both vertices of some square in the grid, will vanish. When $\alpha = \beta$, the entries corresponding to diagonals of a square will also vanish, resulting in a matrix which has the sparsity pattern of the grid adjacency matrix and is very close to the combinatorial Laplacian of the grid (featuring 4 on the diagonal and -1 on the entries corresponding to edges of the grid). In the ASAP case ($\alpha, \beta \neq 0$), $\mathbf{L}_{\alpha,\beta}$ has co-rank 2. In the AAAP case ($\beta = 0$), $\mathbf{L}_{\alpha,\beta}$ is typically significantly rank deficient, with co-rank on the order of the number of boundary vertices. This may be rectified by taking $\alpha = 1$ and β extremely small (on the order of 10^{-3} or 10^{-4}). Fig. 4 shows examples of constrained ASAP warps of a square grid obtained by minimizing $\mathbf{L}_{\alpha,\beta}$ for $\alpha = 1$ and different values of β . For

a sparse 5×5 grid, there is a noticeable difference between the results for the different values of β . For a dense 17×17 grid, there is a striking difference between the AAAP case of very small values of β , and for $\beta = 1$. For larger values of β there is no change. For comparison, when a regular (equilateral) triangular grid of 33 vertices is used, there is no way to obtain the equivalent of the AAAP warp, and the result (Fig. 4 bottom right) is, as expected, somewhat similar to the two-triangle ASAP warp of the square grid mesh ($\beta = 1$), albeit with a conformal higher energy value, indicating that the quad mesh has superior convergence properties.

3.3. General quad

The construction of $\mathbf{S}_{\alpha,\beta}$ for a single square developed in Section 3.1 may be generalized to a general quad Q having geometry vector $\mathbf{q} = (q_j)$. Any affine transformation in the complex plane may be expressed as $A(z) = rz + s\bar{z} + t$, where $z = x + iy$ is a complex number (representing the two coordinates of a planar point), \bar{z} is the complex conjugate of z and r, s, t are complex coefficients. Thus, in the terminology of Section 3.1, the 4×1 column vector \mathbf{a} is orthogonal to the 3×4 matrix \mathbf{Q} stacking the complex coordinate vectors $(z_j, \bar{z}_j, 1)^t, j = 1, \dots, 4$ of the vertices of Q . The matrix $\mathbf{A} = \mathbf{a}\mathbf{a}^*$ is a rank-1 4×4 complex Hermitian semi-positive definite matrix and the quadratic AAAP energy $E_{\alpha,0}$ vanishes on all quads which are affine copies of Q . The ASAP energy is even more simple, as a similar (conformal) transformation in complex numbers is just $S(z) = rz + t$, where r and t are two complex numbers. Thus the 4×1 column vector \mathbf{b} is orthogonal to the 2×4 matrix \mathbf{Q} stacking the complex coordinate vectors $(z_j, 1)^t, j = 1, \dots, 4$ of the vertices of Q , and also to \mathbf{a} . Thus, as in Section 3.1, the ASAP energy matrix $\mathbf{S}_{\alpha,\beta}$ is defined as $\alpha\mathbf{a}\mathbf{a}^* + \beta\mathbf{b}\mathbf{b}^*$ and the matrix $\mathbf{L}_{\alpha,\beta}$ for a mesh of quads may be assembled, as described in Section 3.2, by summing the matrices of the individual quads.

3.4. General triangle

For the special case of a general (not necessarily equilateral) triangle, the number of edges of the polygon is $d = 3$ and the mapping between a source triangle and its target is unique: the affine mapping (which is also harmonic). In this case \mathbf{S} is just the outer product of a single null vector of \mathbf{Q} . It is easy to verify that

$$\mathbf{a} = (e_1, e_2, e_3)^* \quad \text{and} \quad \mathbf{S} = \frac{1}{8A} \mathbf{a} \mathbf{a}^*$$

where e_j are the (complex) edge vectors of the triangle (e_j corresponds to the edge *opposite* vertex q_j) and A is the area of the triangle. The factor $1/8A$ is in order to be compatible with the so-called *cotangent* Laplacian weights of a triangle mesh [PP93], and is obtained by requiring that $\bar{\mathbf{q}}^* \mathbf{S} \bar{\mathbf{q}} = 2A$. Note that adopting just the Procrustes distance of (1) would result in $\mathbf{S}_p = \mathbf{a} \mathbf{a}^* / L$, where L is the sum of squares of the triangle edge lengths, related to the geometry of the triangle, but not the correct quantity. Thus \mathbf{S} and \mathbf{S}_p are related by the so-called *harmonic index* of the triangle [Mus97] – the ratio between L and A . The harmonic index is also $(8 \times)$ the single non-zero eigenvalue of \mathbf{S} (also the trace of \mathbf{S}). For a complete triangle mesh, the resulting \mathbf{L} is identical to the classical ASAP/DNCP energy matrix used in discrete conformal mapping scenarios [LM02; DMA02; MTAD08; LZX*08], containing imaginary values in entries corresponding to boundary edges, representing so-called “natural” boundary conditions. Here too the trace of \mathbf{L} is (up to a constant) equal to the harmonic index of the mesh (the sum of the harmonic indices of the individual triangles). Note again that there are no degrees of freedom in \mathbf{S} , thus a triangle mesh cannot be used for an AAAP deformation.

3.5. Line segment constraints

In general, minimizing the $E_{\alpha,\beta}$ energy function (3) is meaningless without constraints, as the trivial identity (or any global similarity applied to the given square grid G) will have vanishing energy. In the digital photography applications we will consider later, the most useful types of constraints are line segments, each indicating a correspondence between a linear feature in each of the two images, e.g. the corner or silhouette of a building. Given a line segment r_k ($k = 1, \dots, m = \text{number of line segments}$) with endpoint geometry vectors $(\mathbf{r}_{k,0}, \mathbf{r}_{k,1})$ in the source image discretized by the square grid G and its corresponding image $s_k = (\mathbf{s}_{k,0}, \mathbf{s}_{k,1})$ in the target image, the former may be sampled at $l + 1$ points $\mathbf{r}_{k,j} = (1 - j/m)\mathbf{r}_{k,0} + (j/m)\mathbf{r}_{k,1}$, $j = 0, \dots, l$. Each such point $\mathbf{r}_{k,j}$ falls in some quad of G , thus may be described as a linear combination of the four quad vertices (using bilinear coordinates in the case of a square quad, or any other barycentric coordinate scheme, (e.g. [Flo03]): $\mathbf{r}_{k,j} = \mathbf{G}_{k,j} \mathbf{c}_{k,j}$ where $\mathbf{G}_{k,j}$ is the 3×4 (homogeneous) geometry matrix of the relevant quad and $\mathbf{c}_{k,j}$ is a 4×1 (column) vector of real coefficients. There are two possible ways to constrain the linearly sampled $\mathbf{r}_{k,j}$: In the simplest version, they are mapped to their linearly sampled counterparts $\mathbf{s}_{k,j}$ which we call *linearly-sampled line constraints*:

$$\mathbf{H}_{k,j} \mathbf{c}_{k,j} = \mathbf{s}_{k,j}$$

In the more sophisticated version, we constraint the $\mathbf{r}_{k,j}$ to be mapped onto the target line segment, *without* a hard constraint on its precise location. We call the latter *sliding points* or *flexible line constraints*. Assume the line segment s_k lies on the line whose equation is $\mathbf{b}_k(x, y, 1)^t = 0$ for some 1×3 (row) vector of 3 constants \mathbf{b}_k . Then the constraint for $\mathbf{r}_{k,j}$ is

$$\mathbf{b}_k \mathbf{H}_{k,j} \mathbf{c}_{k,j} = 0$$

where $\mathbf{H}_{k,j}$ is the (unknown) 3×4 (homogeneous) geometry matrix of the relevant target quad geometry. These constraints give the m points sampled on r_k the freedom to “slide” to any position along the line described by s_k . This enables a non-linear spacing of the images of the samples of r_k on s_k , which may better describe a perspective image of a line segment. It is good practice, however, to require that the endpoint of r_k map to the endpoints of s_k , i.e. $\mathbf{r}_{k,0}$ maps to $\mathbf{s}_{k,0}$ and $\mathbf{r}_{k,1}$ to $\mathbf{s}_{k,1}$.

Thus the energy deformation problem boils down to the following quadratic minimization problem subject to linear constraints:

$$\begin{aligned} \min_{\mathbf{h}} \quad & \mathbf{h}^* \mathbf{L}_{\alpha,\beta} \mathbf{h} \\ \text{s. t.} \quad & \mathbf{B}^* \mathbf{h} = \mathbf{t} \end{aligned} \quad (4)$$

where each row in the equality constraints $\mathbf{B}^* \mathbf{h} = \mathbf{t}$ is one of the following types

$$\begin{aligned} \mathbf{H}_{k,j} \mathbf{c}_{k,j} &= \mathbf{s}_{k,j} & \text{if } k\text{'th constraint is linearly sampled} \\ \mathbf{b}_k \mathbf{H}_{k,j} \mathbf{c}_{k,j} &= 0 & \text{if } k\text{'th constraint is flexible} \end{aligned}$$

Note that all the linear equalities, including those of the “sliding points”, are non-homogeneous (i.e. have a non-zero rhs). While this formulation does not enforce any particular ordering of the points along the line segment, in practice the ordering of the sample points in the source will be preserved, as is evident in Fig. 5.

The closed form solution to (4) is

$$\mathbf{h} = (\mathbf{L}_{\alpha,\beta}^* \mathbf{L}_{\alpha,\beta} + \mathbf{B} \mathbf{B}^* - \mathbf{L}_{\alpha,\beta}^* \mathbf{B} \mathbf{B}^* \mathbf{L}_{\alpha,\beta})^{-1} \mathbf{B} \mathbf{t} \quad (5)$$

where \mathbf{B}^+ is the pseudo inverse of \mathbf{B} .

3.6. Solving for the warp

Naively solving for the generalized ASAP warp using (5) involves the computation of the inverse of a $n \times n$ matrix. If $\mathbf{L}_{\alpha,\beta}$ were itself invertible, (5) is equivalent to the simpler expression

$$\mathbf{h} = \mathbf{L}_{\alpha,\beta}^{-1} \mathbf{B} (\mathbf{B}^* \mathbf{L}_{\alpha,\beta}^{-1} \mathbf{B})^{-1} \mathbf{t} \quad (6)$$

which may be derived using the method of Lagrange multipliers and has the advantage that once $\mathbf{L}_{\alpha,\beta}^{-1}$ is computed and only the constraint matrix \mathbf{B} changes between instances of the problem, a solution involves computing only the inverse of a much smaller matrix (the so-called *Schur complement* trick). This may be further optimized by pre-factoring the fixed $\mathbf{L}_{\alpha,\beta}$. Although in our scenario $\mathbf{L}_{\alpha,\beta}$ does not have full rank, thus (6) is not directly applicable, Jacobson [Jac14] has shown how to extend the Schur complement trick to deal also with this case.

4. Experimental Results

Many schemes have been proposed to warp 2D images, the most relevant to the applications we have in mind being the moving least-squares (MLS) methods [SMW06]. The affine-based MLS method produces a continuous map of the plane to itself based on point and line segment constraints (which are implicitly linearly sampled within an integral, i.e. flexible lines are not possible in the MLS framework). Essentially each source point is mapped to its image using the affine transformation which best fits the constraints in a weighted least-squares sense, where each constraint is weighted according to its distance from the source point. The same idea can be used to produce a similarity-based MLS, where source points are mapped using best-fitting similarity transformations. In our experiments, we compared our results to both affine-based and similarity-based MLS.

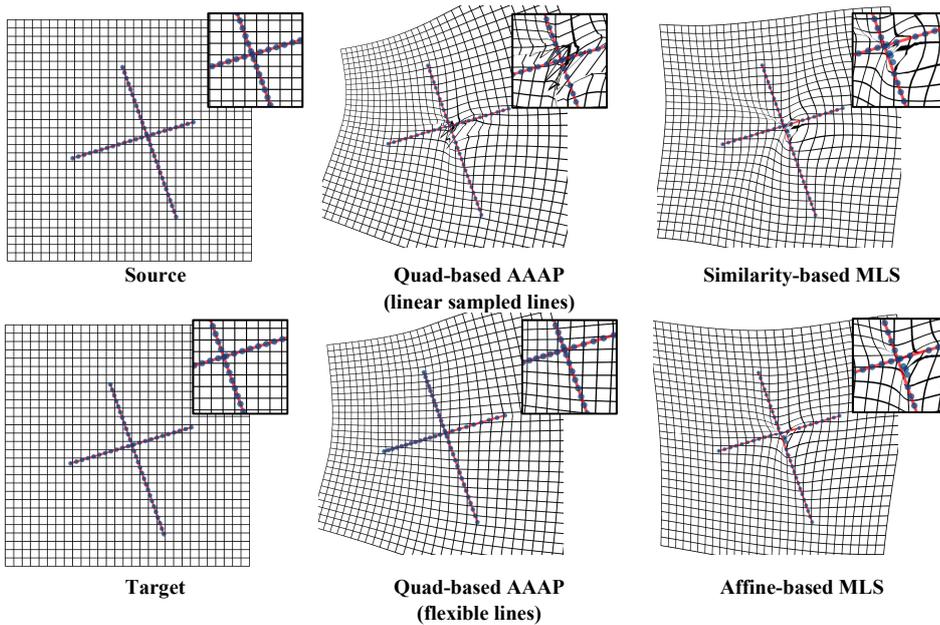


Figure 5: Comparison of warps generated by different methods on very simple input, where one target line constraint is horizontally translated to the left relative to the corresponding source line segment. The target is warped to conform to the line constraints in the source. Insets are zooms into the intersection of the two lines. Blue points are the images (in the warp) of constraint points sampled on the target line segments. Note their non-linear spacing when flexible lines are allowed. Note also how some of these points stray from the target line images in the MLS result, since MLS does not enforce true hard constraints.

To apply AAAP, we discretize the image plane to a square grid. Using the terminology of Section 3.1 and 3.2, we may assume, without loss of generality, that $\alpha = 1$. Slightly abusing the notation, we denote $\mathbf{L}_\beta = \mathbf{L}_{1,\beta}$ (β is the one degree of freedom mentioned in Section 1). In our experiments, we used an ASAP matrix and an AAAP matrix. The ASAP matrix was \mathbf{L}_3 , corresponding to a bilinear mapping of the interiors of the squares. Ideally the AAAP matrix would have been \mathbf{L}_0 , but since the co-rank of \mathbf{L}_0 is much larger than 3 (it is typically around half the number of the grid boundary vertices), many constraints are needed to produce a non-trivial solution. To avoid numerical issues we “regularize” by using a biased matrix \mathbf{L}_β , where $\beta \in [0.001, 0.01]$. This matrix has co-rank 2.

To compare the different methods, we start with some simple synthetic inputs and visualize the resulting warps by its effect on an image of a square texture map. The objective is to deform the target 2D grid to conform to the constraints in the source 2D grid. Fig. 5 shows this for a very simple input of two line constraints, where one target line segment is identical to its source counterpart and the other is translated horizontally relative to its source. The source line segment is sampled by one point per grid square on the average. Imposing point constraints by linearly sampling the line segments produces bad results as obviously these constraints along both lines will conflict in the intersection region. As a result, both affine-based and similarity-based MLS (having implicit linear sampling) and AAAP contort the grid very awkwardly in order to satisfy the constraints. In contrast, “sliding” line constraints (also called “flexible lines” in the figures) with AAAP produce much better results, close to a piecewise-affine map. In all our experimental results, AAAP and ASAP used a square grid where each square contained 5×5 image pixels (texels). This grid used to discretize the plane is not to be confused with the square grid visible in this example,

which is a much coarser grid drawn in the input images in order to better visualize the warp. The MLS method weights the constraints by $1/d^2$, where d is the distance of the constraint from the warped pixel. The warped image was rendered by texture mapping the target image onto the deformed quad mesh using bilinear mapping. Notice also in the zoom insets that the sampled point constraints are actually not perfectly satisfied by MLS, since the formulation imposes hard constraints as infinitely-weighted soft constraints, and this might not be good enough in extreme settings such as this.

4.1. Perspective Warp

A fundamental assumption that we rely on in both the photographic applications treated in this paper is that the AAAP warp provides a good approximation of the perspective (homography) warp that is typically present in the images of planar surfaces in photographs. We verify this in Fig. 6, where we show the warp of a square region (32×32 square grid with 20×20 texels per quad) to a trapezoidal region using a combination of variations of the generalized ASAP method: linear or flexible boundary conditions and $\beta = 10^{-4}, 1$. For linear boundary conditions, the result for both values of β was quite close to a bilinear mapping of the square domain. For flexible boundary conditions, $\beta = 3$ yielded a conformal-type mapping (with significant “bending” of the straight checkerboard lines), whereas the AAAP case ($\beta = 10^{-4}$) yielded a close approximation to the perspective warp (a homography). Note how the vertical spacing of the checkerboard squares is not uniform.

For a real-world application, Fig. 2 shows a photograph of a kitchen scene, and three possible warps, each attempting to modify the perspective of the camera. The first is that generated by the system of Carroll et al [CAA10], with a variety of constraints. The second is

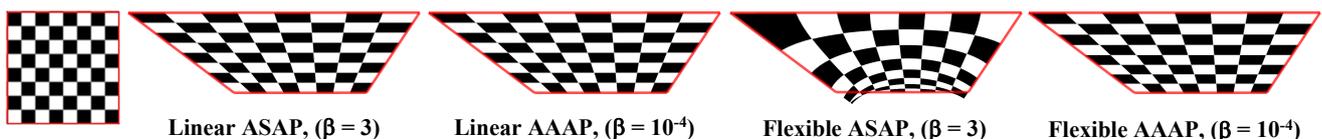


Figure 6: Attempt to reproduce the perspective warp (homography) using generalized ASAP. Linear boundary conditions yield a result close to the bilinear map. With flexible boundary conditions, ASAP ($\beta = 3$) yields a conformal-type map, while AAAP ($\beta = 10^{-4}$) yields a close approximation to the perspective warp.

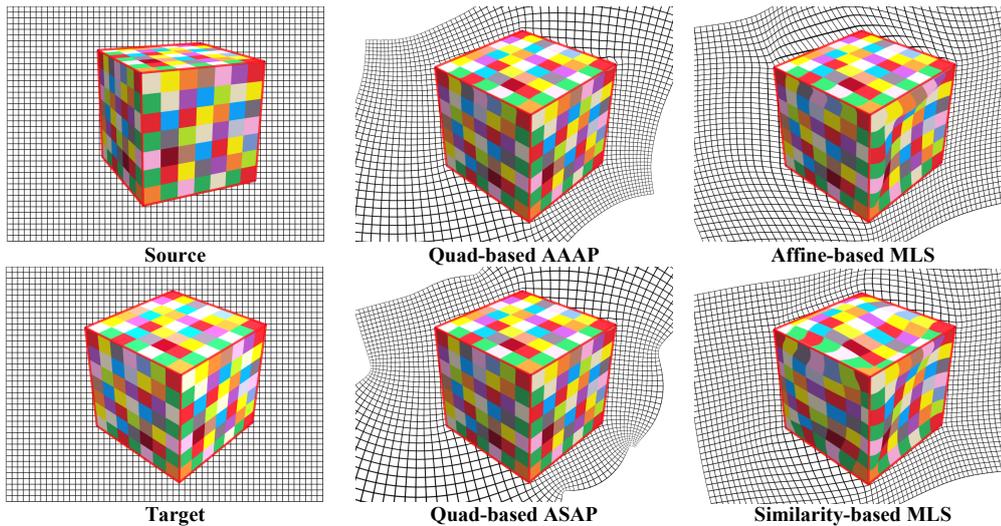


Figure 7: Comparison of different warps of an image of a 3D cube with nine line constraints. The source image is warped to conform to the constraints in the target image. Note the excellent results of the quad-based ASAP and AAAP, although the AAAP result has a slight artifact along the top edge of the leftmost face.

our AAAP warp, fed only a small number of line segment constraints. The resulting warp is very similar to that of Carrol et al, at a fraction of the computational cost. In contrast, the affine-based MLS warp [SMW06], which is a continuous warp based on the same linear constraints, contains some distortions and artifacts, most notably that unconstrained vertical lines do not stay vertical, or even straight. Our implementation of MLS follows Appendix B of [SMW06], using the closed-form (integrated) line segment constraints, implicitly assuming a linear parametrization of the corresponding line segments.

4.2. Re-photography

Fig. 7 shows a synthetic re-photography scenario, comparing the warps generated on a representative input pair where the perspective on a rendering of a simple textured 3D cube changes between images. The line constraints delineate the boundaries of the three visible faces. Line segments terminating at a common vertex are “snapped” to a common point to make sure they coincide perfectly. Quad-based ASAP produces an excellent result, albeit with some artifacts along the cube edges. Quad-based AAAP produces slightly less perfect results. MLS seems to produce the worst results, with noticeable artifacts.

A real-world re-photography example may be seen in Fig. 1. The left image depicts a vintage photo taken in 1911 in New York City, and the center image a photo taken of the same scene in 2014. Corresponding line segment features are marked in red on both photos. Note the difference in perspective, illustrated in the buildings common to both photos. The cream-colored ornate building in the center background appears larger in the vintage photo, yet the brown subway station in the center foreground is smaller. The right image shows an AAAP warp of the contemporary photo which conforms to the corresponding line segment constraints marked in red on the two input photos, in lieu of a re-photograph of the vintage photo. The quality of the result is illustrated by combining the warped contemporary photo with the vintage one. Now the two buildings common to both photos aligned well.

Fig. 8 shows more real-world examples of correcting a re-photography attempt with user-defined constraints, again comparing the different methods. As expected, line segments capture image features well and are a convenient way to guide the warp. MLS seems to respect the constraints in the regions close to them, but produce

artifacts in more distant regions, resulting in image elements “bending” in an unnatural way. ASAP has a similar problem in regions distant from the constraints, but even within a large polygonal area enclosed by constraints, it introduces “bends”, as evident in the columns of the building in the first image. There is almost no difference between triangle-based and quad-based ASAP, as expected. In contrast, quad-based AAAP results in a natural deformation (remember that AAAP is impossible on a triangle mesh). In all these examples, involving images of approximately 1000×1000 pixels, the warp is solved in real-time on a medium-range Windows laptop. We found that a modest resolution, e.g. $61 \times 61 = 3,721$ grid points for a square image is sufficient to produce good results. All quad-based deformations use bilinear mapping for the quad interiors.

In our software implementation, we concentrated mostly on the re-photography application, developing an interactive system, where the user sees the vintage and contemporary images side-by-side and can add/delete and edit corresponding line segments in both images. The contemporary image may then be warped accordingly and saved. The warp is fast enough to be done in real-time as the user adjusts the constraints, but we have sometimes found that it is more convenient to just edit the constraints on both images, and then to see the resulting warp of the contemporary image at the push of a button. A sample session may be seen in the video accompanying this paper. More re-photography results may be seen in the additional material accompanying this paper.

5. Summary and Conclusion

We have described a general 2D deformation method of square unit grids, based on quadratic energy functions, which produces image warps suitable for a number of digital photography applications: a change of camera perspective and re-photography – the offline correction of a contemporary photo to coincide better with a vintage photo taken under similar conditions. Our experimental results show that our AAAP warp on a square grid is an effective and efficient method for these applications, guided by manual identification of common feature line segments. Some trial and error is required from the user in selecting and positioning the key feature line segments, but our experience shows that no more than a handful of line segments positioned within a few minutes of user interaction is needed to obtain a good result on typical inputs. Although these warps do not guarantee that the mapping is injective (i.e. does

not contain “flips”), this can usually be avoided by imposing appropriate line segment constraints.

The warp of a unit square grid is just one member of a much broader family of warps, all of which may be generated from a polygonal discretization of the plane. The resulting warp is computed as the minimum of a quadratic energy function, tailored to achieve certain effects subject to linear constraints, where the system matrix of the quadratic form is a sparse matrix having a structure similar to that of the graph Laplacian. An additional possible direction, which we have not explored here, is to weight differently the ASAP energy of each polygon in the discretization of the image plane, possibly by image saliency (so-called *content-aware*). This could possibly generate more interesting warps with less visible artifacts. It would also be interesting to characterize the matrices and maps derived from some common tilings of the plane, such as regular hexagonal tilings or Delaunay triangulations.

Acknowledgements

R. Chen has been supported by the Max Planck Center for Visual Computing and Communication.

References

- [ACL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. *Proc. ACM SIGGRAPH* (2000), 157-164.
- [AW11] ALEXA M., WARDETZKY M.: Discrete Laplacians on general polygonal meshes. *ACM Transactions on Graphics* 30, 4 (*Proc. ACM SIGGRAPH*) (2011), 102:1-102:10.
- [BAD10] BAE S., AGARWALA A., DURAND F.: Computational re-photography. *Proc. ACM SIGGRAPH*, 29, 3 (2010).
- [Boo89] BOOKSTEIN F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. PAMI*, 11, 6 (1989), 567-585.
- [CAA10] CARROLL R., AGARWALA M., AGARWALA A.: Image warps for artistic perspective manipulation. *Proc. ACM SIGGRAPH* (2010).
- [DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, 21, 3 (2002), 209-218.
- [Flo03] FLOATER M.: Mean-value coordinates. *Computer Aided Geometric Design*, 20, 1 (2003), 19-27.
- [FH05] FLOATER M., HORMANN K.: Surface parameterization: A tutorial and survey. In: *Advances in Multiresolution for Geometric Modelling* (Dodgson, Floater and Sabin, Eds.), pp. 157-186, Springer, (2005).
- [GSC06] GAL R., SORKINE O., COHEN-OR D.: Feature-aware texturing. *Proc. EG Symp. on Rendering* (2006).
- [HZ04] HARTLEY R., ZISSERMAN A.: *Multiple View Geometry* (2nd Ed.), Cambridge University Press, 2004.
- [IMH05] IGARASHI T., MOSCOVICH T., HUGHES J.F.: As-rigid-as-possible shape manipulation. *Proc. ACM SIGGRAPH*, 24, 3 (2005), 1134-1141.
- [JBPS11] JACOBSON A., BARAN I., POPOVIC J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *Proc. ACM SIGGRAPH* (2011).
- [Jac14] JACOBSON A.: Schur complement trick for positive semi-definite energies. *Technical Report*, Columbia University, 2014.
- [JMD*07] JOSHI P., MEYER M., DE ROSE T., GREEN B., SANOCKI T.: Harmonic coordinates for character articulation. *Proc. ACM SIGGRAPH*, 26, 3 (2007).
- [KFG09] KARNI Z., FREEDMAN D., GOTSMAN C.: Energy-based image deformation. *Computer Graphics Forum*, 28, 5 (2009), 1257-1268.
- [KWS*13] KAUFMANN P., WANG O., SORKINE-HORNUNG A., SORKINE-HORNUNG O., SMOLIC A., GROSS M.: Finite element image warping. *Proc. Eurographics* (2013).
- [Lev04] LEVERE D.: *New York Changing: Revisiting Berenice Abbott's New York*. Princeton Architectural Press, 2004.
- [LLC11] LEE K.-T., LUO S.-J., CHEN B.-Y.: Re-photography using image collections. *Proc. Pacific Graphics* (2011).
- [LLM*11] LIN W.-Y., LIU S., MATSUSHITA Y., NG T.-T., CHEONG L.-F.: Smoothly varying affine stitching. *Proc. CVPR* (2011).
- [LM02] LEVY B., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *Proc. ACM SIGGRAPH*, 21, 3 (2002), 362-371.
- [LZX*08] LIU L., ZHANG L., XU Y., GOTSMAN C., GORTLER S.J.: A local/global approach to mesh parameterization. *Computer Graphics Forum*, 27, 5 (2008), 1495-1504.
- [MRT12] MARTINEZ ESTURO J., ROESSL C., THEISEL H.: Generalized metric energies for continuous shape deformation. *Springer LNCS (Proc. Curves and Surfaces)* 8177, 1 (2012).
- [MTAD08] MULLEN P., TONG Y., ALLIEZ P., DESBRUN M.: Spectral conformal parameterization. *Computer Graphics Forum*, 27, 5 (2008), 1487-1494.
- [Mus97] MUSIN O.: Properties of the Delaunay triangulation. *Proc. Symp. on Comp. Geom. (SoCG)*, (1997), 424-426.
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experiment. Math.*, 2, 1 (1993), 15-36.
- [PWS12] PANOZZO D., WEBER O., SORKINE O.: Robust image re-targeting via axis-aligned deformation. *Computer Graphics Forum*, 31, 2 (2012), (*Proc. Eurographics*).
- [Que05] QUESNEY D.: *Retour a Paris: Identical shots, a hundred years apart*. Parigramme, (2005).
- [SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *Proc. ACM SIGGRAPH* (2006), 533-540.
- [WBGH11] WEBER O., BEN-CHEN M., GOTSMAN C., HORMANN K.: A complex view of barycentric mappings. *Computer Graphics Forum (Proc. Symp. Geometry Proc.)*, 30, 5 (2011), 1533-1542.
- [WTSL08] WANG Y.-S., TAI C.-L., SORKINE O., LEE T.-Y.: Optimized scale-and-stretch for image resizing. *Proc. SIGGRAPH Asia*, 27, 5 (2008), 118.
- [YJHS12] YUCER K., JACOBSON A., HORNUNG A., SORKINE O.: Transfusive image manipulation. *Proc. SIGGRAPH Asia*, 31, 6 (2012).
- [ZCBS13] ZARAGOZA J., CHIN T.-J., BROWN M.S., SUTER D.: As-projective-as-possible image stitching with moving DLT. *Proc. CVPR* (2013).
- [ZCHM09] ZHANG G.-X., CHENG M.-M., HU S.-M., MARTIN R.: A Shape-preserving approach to image resizing. *Computer Graphics Forum (Proc. Pacific Graphics)*, 28, 7 (2009).



Figure 8: Comparison of re-photography results using our quad-based AAAP algorithm with flexible lines vs. affine-based MLS and quad-based and triangle-based ASAP. The user-defined line constraints are marked in red. Vintage photographs courtesy of www.nycvintageimages.com. All quad-based deformations use bilinear mapping for the quad interiors. **(Top)** Note how the church and bicycle rider are “fat” and the building with arches seems to bend in the MLS and ASAP results. **(Bottom)** Note the distorted Flatiron building in the MLS result, due to the implied linear sampling of the constraint lines. The ASAP result demonstrates the distortion in regions distant from the constraints.

Appendix: Conformal energy of a barycentric mapping of the unit square

Assume that the mapping f of the interior of the unit square $\mathbf{s} = (s_j)^t = (0, 1, 1 + i, i)^t$ with vertices in counter-clockwise order to the interior of an arbitrary quad $\mathbf{q} = (q_j)^t$ is by a barycentric coordinate scheme, namely, if $z = \sum_{j=1,\dots,4} B_j(z) s_j$, then $f(z) = \sum_{j=1,\dots,4} B_j(z) q_j$. Make also the (standard) assumption that f maps the edges of \mathbf{s} linearly to the edges of \mathbf{q} . By definition, the conformal energy of f is:

$$E_c(f) = 2 \iint_S \left| \frac{\partial f}{\partial \bar{z}} \right|^2 dz \wedge d\bar{z}$$

($dz \wedge d\bar{z}$ is just the standard area differential in complex form).

Now, in matrix form, $f(z) = \mathbf{B}(z)^t \mathbf{q}$, thus $\frac{\partial f}{\partial \bar{z}} = \frac{\partial \mathbf{B}(z)^t}{\partial \bar{z}} \mathbf{q}$,

$$\left| \frac{\partial f}{\partial \bar{z}} \right|^2 = \mathbf{q}^* \left[\frac{\partial \mathbf{B}(z)}{\partial z} \frac{\partial \mathbf{B}(z)^t}{\partial \bar{z}} \right] \mathbf{q}$$

and $E_c(f) = \mathbf{q}^* \mathbf{S} \mathbf{q}$, where $\mathbf{S}_{jk} = 2 \iint_S \left(\frac{\partial B_j(z)}{\partial z} \right) \left(\frac{\partial B_k(z)}{\partial \bar{z}} \right) dz \wedge d\bar{z}$.

The integral in \mathbf{S}_{jk} is a double integral over a parameterization of the unit square $(u, v) \in [0, 1]^2$, where $u = (z + \bar{z})/2$, $v = (z - \bar{z})/2i$.

Let us analyze some special barycentric maps. The first is when \mathbf{s} is split into two triangles, and each is mapped affinely to the corresponding triangle of \mathbf{q} (see Fig. 9 (top)). In this map, one of the four barycentric coordinates is always zero. For points in the first triangle having vertices $(0, 1, i)$, we have $\mathbf{B}(z) = (1 - u - v, u, 0, v)^t = \frac{1}{2}(2 + (i - 1)z - (i + 1)\bar{z}, z + \bar{z}, 0, -iz + i\bar{z})^t$, hence $\frac{\partial \mathbf{B}(z)}{\partial \bar{z}} = \frac{1}{2}(-i - 1, 1, 0, i)^t$ and the matrix \mathbf{S}_1 is

$$\mathbf{S}_1 = \frac{1}{4}(i - 1, 1, 0, -i)^t(-i - 1, 1, 0, i)$$

For the second triangle having vertices $(1, 1 + i, i)$, we have $\mathbf{B}(z) = (0, 1 - v, u + v - 1, 1 - u)^t = \frac{1}{2}(0, 2 + iz - i\bar{z}, (1 - i)z + (1 + i)\bar{z} - 2, 2 - z - \bar{z})^t$, hence $\frac{\partial \mathbf{B}(z)}{\partial \bar{z}} = \frac{1}{2}(0, -i, i + 1, -1)$ and the matrix \mathbf{S}_2 is

$$\mathbf{S}_2 = \frac{1}{4}(0, i, -i + 1, -1)^t(0, -i, i + 1, -1)$$

Thus the conformal energy matrix for this mapping is

$$\mathbf{S}_{tri} = \mathbf{S}_1 + \mathbf{S}_2 = \frac{1}{4} \begin{pmatrix} 2 & -1 + i & 0 & -1 - i \\ -1 - i & 2 & -1 + i & 0 \\ 0 & -1 - i & 2 & -1 + i \\ -1 + i & 0 & -1 - i & 2 \end{pmatrix}$$

which is $S_{\frac{1}{4}4}$ of Section 3.1.

The second barycentric mapping is the bilinear map between \mathbf{s} and \mathbf{q} , which is also harmonic, and happens to also coincide with the discrete harmonic (cotangent) and Wachspress barycentric coordinate functions [WBGH11] (see Fig. 9 (bottom)). In this case the barycentric coordinate functions are

$$\mathbf{B}(z) = ((1 - u)(1 - v), u(1 - v), uv, (1 - u)v)^t,$$

and

$$\frac{\partial \mathbf{B}(z)}{\partial \bar{z}} = \frac{1}{2} \begin{pmatrix} i\bar{z} - 1 - i \\ 1 - i\bar{z} \\ i\bar{z} \\ -i\bar{z} + i \end{pmatrix}$$

resulting in

$$\mathbf{S}_{bil} = \frac{1}{12} \begin{pmatrix} 4 & -1 + 3i & -2 & -1 - 3i \\ -1 - 3i & 4 & -1 + 3i & -2 \\ -2 & -1 - 3i & 4 & -1 + 3i \\ -1 + 3i & -2 & -1 - 3i & 4 \end{pmatrix}$$

which is $S_{\frac{1}{12}4}$ of Section 3.1

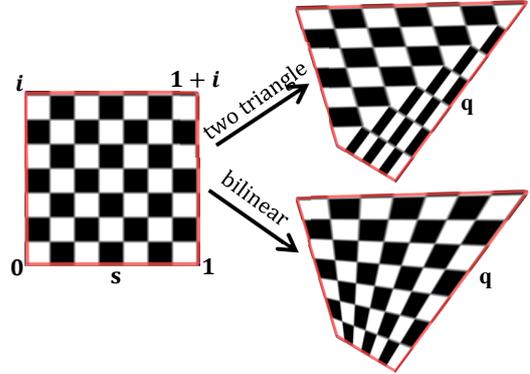


Figure 9: Barycentric mappings of the square \mathbf{s} to a quad \mathbf{q} . The two-triangle mapping (top) subdivides \mathbf{s} and \mathbf{q} into two corresponding triangles and maps each corresponding pair of triangles with the unique affine map between them. Note the C^1 discontinuity along the diagonal. The conformal energy of this mapping is $\mathbf{q}^* \mathbf{S}_{tri} \mathbf{q} = 20.4$. The bilinear mapping (bottom) is C^1 and its conformal energy is $\mathbf{q}^* \mathbf{S}_{bil} \mathbf{q} = 18.4$.

The third barycentric map is based on the mean-value coordinates [Flo03], which, in complex form [WBGH11] are

$$\mathbf{B}(z) = \begin{pmatrix} \frac{|z|(1 - u - v) + |z - 1|u + |z - i|v}{uv} \\ \frac{|z|(1 - u) + |z - 1 - i|v + |z - 1|(u - v)}{(1 - u)v} \\ \frac{|z - 1 - i|(v + u - 1) + |z - 1|(1 - v) + |z - i|(1 - u)}{(1 - u)(1 - v)} \\ \frac{|z|(1 - v) + |z - 1 - i|u + |z - i|(v - u)}{u(1 - v)} \end{pmatrix}$$

Symbolic manipulation followed by numeric integration yields

$$\mathbf{S}_{mv} = \begin{pmatrix} \frac{410}{1222} & -\frac{62}{725} + \frac{i}{4} & -\frac{88}{535} & -\frac{62}{725} - \frac{i}{4} \\ \frac{62}{725} - \frac{i}{4} & \frac{410}{1222} & -\frac{62}{725} + \frac{i}{4} & -\frac{88}{535} \\ -\frac{88}{535} & -\frac{62}{725} - \frac{i}{4} & \frac{410}{1222} & -\frac{62}{725} + \frac{i}{4} \\ -\frac{62}{725} + \frac{i}{4} & -\frac{88}{535} & -\frac{62}{725} - \frac{i}{4} & \frac{410}{1222} \end{pmatrix}$$

It is interesting to note that all these ASAP matrices have the following common circulant structure:

$$(\alpha + \beta, -\alpha + i\beta, \alpha - \beta, -\alpha - i\beta)$$

with $\beta = \frac{1}{4}(\alpha$ and β have the same meaning as in Section 3.1). The matrices are rank-2 Hermitian with eigenvalues $(0, 0, 4\alpha, 4\beta)$ and common (Fourier) eigenbasis.

Since $\alpha(S_{tri}) = \frac{1}{4}$, $\alpha(S_{bil}) = \frac{1}{12} = 0.0833$ and $\alpha(S_{mv}) = \frac{62}{725} = 0.0855$, this indicates that the bilinear mapping is the most conformal (as expected, since it is harmonic), followed closely by the mean-value mapping, and then distantly by the two-triangle mapping.