ACM SIGGRAPH

SIS Conference Reporting

210 - Submission Reviews - By Person

Generated: March 11, 2014 01:40:25 EDT

Question	Response
Id:	papers_0210
Title:	BendFields: Regularized Curvature Fields from Concept Sketches
Reviewer #69:	
1) Description	The paper proposes a method to compute 3D-like shading for bitmap, hand-drawn concept sketches that contain cross-shading curves that originate in curvature lines. The input/output are in principle similar to the ones of the CrossShade paper, but it is not required that the input is vector-graphics. The underlying machinery to create the shading works as follows: the method computes a 3D frame for each pixel of the image, where the tangents are principle directions and the third vector is the surface normal, all coming from an imagined 3D surface depicted in the sketch. To compute this frame field, first a projected 2D field is computed, which corresponds to the principle directions orthographically projected onto the image plane. This 2D field is represented by a non-orthogonal cross field. It is computed by interpolating from the directions estimated from the input sketches and then optimizing a proposed non-linear energy. Then, the 3D frame field is optimized using the 2D field as a kind of initial guess. The main technical contribution of the paper is a proposal how to represent non-orthogonal cross fields and interpolate them, using an extension of the MIQ (mixed-integer quadrangulation) machinery. The second contribution is the energy functional used to interpolate regularized curvature lines. On the practical side, the main application is shading bitmap sketches, but further and broader applications may exist in the geometry processing community, in particular for surface quadrangulation. On the theoretical side, the paper makes an important contribution to extending the representation and smoothness formulation of cross fields from orthogonal to non-orthogonal ones.
2) Clarity of Exposition	The approach is motivated very well in the intro, and the overview section gives a good overall idea of how the method works. That said, the "meat" of the paper (Sections 5-6) will not be very easy to understand for people who are not well familiar with line/cross fields and MIQ, as well as differential geometry. Here I list the comments and questions I had as I was reading the paper. The line numbering is broken in the PDF around equations, unfortunately (there is a LaTeX fix for that). - Line 168 and later in the text: I think a better term for "harmonic energy" would be "Dirichlet energy". - Line 194: "provides" -> provide - Line 257: Do you mean geodesic torsion tau_g or geodesic curvature kappa_g? - Line 259: "designer" -> designers - After line 282, in the paragraph "Regularized curvature cross-field": you write "the generalization to a unit-length tangent vector field T, ..., is E(T, alpha) = ...". This is confusing because E is an energy, not a vector field. Please rephrase to explain that you are interested in T that minimizes E - Notation complaint: I found it confusing that the regularization parameter alpha always appears as one of the arguments in the energy E. You do not optimize for alpha but fix it as a constant, and you do optimize for the other arguments (T, U, V, u, v, etc.) so it is better to separate. The alpha can be a super- or subscript, or omitted altogether. - In the same vein, in the paragraph "2D projection of curvature cross-field" you use notation E(1) to say that alpha is fixed to the value of 1. This is sloppy notation and should be reworked. - Throughout the technical description it was not clear over which 2D (and 3D) domain you optimize, are the boundaries of the sketch somehow identified and are there any boundary conditions. Much later in the results it is mentioned that the frame field is manually masked for pixels that do not belong to the sketched shape - from this I assume that the frame field computation is done for the entire input image. This seems a waste of resources and possibly even wrong (I come back to it later); at the minimum this should be mentioned much earlier in the text when you set up the optimization. - Page 6 top: "Furthermore, for a subset of pixel" -> pixels - Sections 5-6: one thing that was quite difficult to grasp is when are u and v really cross fields (unit-length) and when do they get a variable length. I figured that throughout the cross-field interpolation unit length is kept because the representation with the angles doesn't even contain any length information. However later, after the "harmonic" interpolation, there is another nonlinear optimization to minimize the E energy (after line 421), where I believe the lengths of u, v, are allowed to vary, and u and v are now represented by x-y, not angles. Is this correct?? It would be very helpful if at all mentionings of energy minimization you also explicitly listed ALL the constraints, such as unit length for U, V, N, unit length of u,v (or lack of...), orthogonality (or lack of), etc. Currently this is done rather sporadically and not systematically everywhere. - Line 425: "to flat" -> to be flat - Line 511: "bitmap" -> bitmaps - Figure 12: it is not clear what are the constraints that you interpolate here. The silhouette sketch is drawn in black, which signifies feature curves throughout the paper who do not participate in the curvature line interpolation. There needs to be at least one curvature line sketch (red), no? The corresponding text in lines 529-531 is not explaining this either. - The Appendix is traditionally placed after the bibliography in SIGGRAPH papers. - The ordering of the figures could be optimized, and at least made consecutive with the page ordering. - Figure 11: The input to your algorithm and to CrossShade seems different beyond just vector/raster graphics and annotation: on the handle belt of the bag your system gets an extra crossing line, and the armchair also has an extra vertical line along its back. Why not use exactly the same input up to annotation (simply rasterize the input of CrossShade to use with your method)? After all, the paper is not really about handling problems of raster graphics, and so I think that using very clean input for the comparison to CrossShade is perfectly fine, as long as identical features are represented.
3) Quality of References	The references seems fine. In the first paragraph, since you are talking about sketch inflation, it is maybe worth citing 3D inflation works as well (not just height fields), such as Teddy and FiberMesh. The comparison to CrossShade should make it more clear what is the difference in terms of the algorithm, not the input (in the end, the pixel- or vector-graphics distinction is rather orthogonal to the contribution of this submission). It was not very clear from the paper what are the technical advantages of the new work and the main technological differences.
Submission Information:	

Question	Response
	In line 153 it is maybe better to cite the STAR on quad meshing rather than just Alliez03 and MIQ.
4) Reproducibility	Reproducing this work will be technically very difficult, since nonlinear solvers and mixed-integer solvers are involved, as well as highly non-trivial mathematics and numerics. It is a shame that the submission does not contain any code, executables, not even data (input images, output frames). The MIQ framework is not fully publicly available, as far as I know, which makes reimplementation even harder. That said, it is likely possible if a skilled postdoc or researcher is assigned with the task and invests about half a year to a year; I believe the necessary details are provided in the paper. One practical detail that is missing is the bilateral filtering parameters (section 4), as well as all the stopping criteria and other parameters of the other solvers mentioned in the paper (hierarchical solvers, IPOPT etc.). Releasing the code for academic research and comparison purposes would be highly beneficial to our community.
6) Rating	3.9
8) Explanation of Rating	In essence I think this paper is really exciting. The results are beautiful and the mathematical idea is quite nontrivial yet elegant. It is surprising that one can solve such an ill-posed problem as shading and parametrization reconstruction from sparse 2D sketches, but with the right assumptions this paper proves it possible. I like the regularized curvature lines formulations very much. The extension of cross-field interpolation to non-orthogonal cross fields is an interesting contribution on its own, that could be relevant to quad meshing and globally smooth surface parameterization. Hence I think this paper should be accepted for publication after appropriate editing. I have several concerns and suggestions regarding the technical exposition, as detailed above, which the paper should address. Beyond that, still many questions remain (which is a sign that there can be lots of follow-up work here!). Firstly, I wonder whether unit-length cross-fields are the right object to represent projected curvature lines. Allowing varying length directly, or varying "density" would make for a more direct representation of parallel-projected principle direction vectors. It is perhaps challenging to infer the lengths of the cross vectors from the sparse sketches, but they maybe could be estimated from the density of the sketched parallel curves? How the lengths eventually emerge in the second 2D optimization step is somewhat mysterious to me despite several readings of the paper. Second question is why are the black sketches (silhouettes, sharp features) ignored completely. It seems that the silhouette information could provide useful data for the frame optimization - for example, the silhouettes give some information about the surface normals. Currently, if I understood correctly, the frame field is optimized for the entire plane. This seems somewhat counter-intuitive. The boundary conditions provided by the black pixels could be potentially complicated but perhaps they would constrain the problem better? Then I was also wondering how is the user instructed to choose between black (silhouette) and red (curvature line) annotation. Sometimes the red is used for the boundary of the shape (e.g. the back bottom edge of the chair in Figure 10), sometimes it is not. I do not know if designers can be expected to know the mathematical difference between the two kinds of lines. What would be the "naive user" instructions? Related to that, I believe one of the limitations (or let's say, delicate points) not discussed in the paper is the necessary amount of input to make the problem sufficiently well defined. For instance, on the handle of bag in Figure 11, the current algorithm takes more cross-sketches than CrossShade; for the human observer none of these (or maybe just one) would be necessary to understand that the shape is a flat, bended belt. How does the amount of input curves affect the optimization and the result? It would be interesting to see a progression with increasing amount of cross-sketched curves added. Finally, as mentioned before, the comparison to CrossShade should use "harmonized" inputs if possible. A deeper discussion of the mathematical differences would be appreciated, apart from just the differences of input. Also, more results of the current technique would be great to see, including detailed timing information for each one. A smaller question: why are the sketched directions used as soft constraints (with a high weight) rather than incorporated (substituted) directly as hard constraints? Doesn't the high weight on the least-squares constraints make the numerics less stable?

Reviewer #31:

Question	Response
1) Description	This paper describes a method for assigning shape-driven shading to the sketches of concept shape sketches. The main contribution is a method for extracting a dense principal curvature field given sparse curve field input in the form of sketched curves. This extracted principal curvature field is then used to parameterize and shade the input sketch. I'm not entirely convinced that the proposed method for extrapolating curvature lines is better than standard methods that rely on third order shape information (I've listed some missing references in 'Quality of References' section). I think the method proposed in the paper will work well for the target application of the paper. I'm a little bit on the fence with the scope of this paper. While I can clearly see the 3D shape inferred by the shaded sketches, I can also see the 3D shape inferred by the input, non-shaded sketch. The 3D shading does not seem to disambiguate the input (I suspect the disambiguation of the intended shape is handled by the user interface). Therefore, I understand this paper's application to be to visually enhance the look of the sketch without conveying any significant new information. That is fine for Siggraph, but it does lower the scope somewhat. I would have been more enthusiastic to support this paper if we had an accompanying user study comparing the shape information inferred from sketches with and without shading.
2) Clarity of Exposition	The exposition is mostly clear. Some suggestions: General comment: on page 4 the authors introduce several energy functionals, each one taking different number of inputs, and each one labeled 'E'. It would be easier if a subscript was added to each 'E' differentiating it from the others. line 205: I'm assuming 'I' here refers to the vector data stored per pixel in the image? (e.g. RGB)? line 215: please give some more details on the bilateral filter used...size of the filter kernel, etc. line 423: the comment on the nonlinear optimization was interesting. How confident are the authors that they found the actual (local) minimum v/s just a result that looked good? My experience with repeated application of linear approximants of non-linear functionals has not been very positive. It is hard to know if one is indeed at the minimum in general. This point is not too significant because I suspect even a few iterations of the energy minimization will yield 'good enough' results, and additional iterations may not visually change the result significantly.

Submission Information:

Question	Response
	line 453: I think the paper needs more details for the user interface section. I did not understand how the input was split up into charts and how each chart/patch boundary was assigned the 3D normals/curvature lines that could be interpolated.
3) Quality of References	I think the main missing references are in the area of 3rd order surface interrogation for computing lines of curvature. I would have liked to see a discussion of whether the proposed system will be able to reproduce Darboux's 'star', 'monstar', and 'lemon' curvature patterns at umbilics --- I don't think it can because of the input being split up into charts. A good starting reference that shows how to interpolate curvature lines: Maekawa, T., Wolter, F., Patrikalakis, N., 1996. Umbilics and lines of curvature for shape interrogation. Computer-Aided Geometric Design 13 (2), 133–161. This describes the three pattern types introduced by Darboux: Berry, M., Hannay, J., 1977. Umbilic points on Gaussian random surfaces. Journal of Physics A: Mathematical and General 10 (11), 1809–1821. This provides a more intuitive explanation of when the above patterns are created: Pushkar Joshi, Carlo H. Séquin: An intuitive explanation of third-order surface behavior. Computer Aided Geometric Design 27(2): 150-161 (2010)
4) Reproducibility	The main missing information is in the user interface subsection of Section 7. I would like the exact steps described so that the input for the algorithm is clear. For example, what happens if there are missing or discontinuity lines in the sketch? Must the discontinuity lines enclose a closed region?
6) Rating	3.25
8) Explanation of Rating	As I described above, I'm a little on the fence with the application of this paper. The proposed method is rather involved, requiring number of different numerical optimizations (to be fair, the problem is rather difficult). However, the application seems to be mostly 'eye candy' over the input sketch. That, and the missing user survey advocating the 3D shading is the main reason I'm not more enthusiastic about endorsing this paper. The paper is well written and is mostly sound. The method for extrapolating continuity lines seems a little ad-hoc, when the standard method of 3rd-order analysis would have yielded similar results.

Reviewer #41:

Question	Response
1) Description	The paper presents a new approach to computation of approximate normal fields from hand-drawn concept sketches which has applications in automatic generation of 3D-like shading or 3D surface approximation. As compared to previous techniques it does not require vector input while is still able to better preserve curvature of input lines. The method is based on sound mathematical foundations and presented results look very convincing. There are only two practical limitations: (1) the computational overhead which makes the method unsuitable for interactive applications and (2) limited control over the normal field shape close to region boundaries.
2) Clarity of Exposition	The paper is well written and polished. The only "drawback" I see is that technical part would probably be hard to understand for novice readers as it requires deep understanding of used mathematical principles. I spent quite a long time to understand all technical details (and I have to admit I wasn't fully successful). To alleviate this I'd propose to start from less complex things and proceed towards full formulation. This endeavour is already visible in the paper, but I'd try to improve it, e.g., by making stronger connection to Lumo [Johnston 2002] which is well known to NPR community and it was already shown, e.g., in [Winnemoller et al. 2009] or in TexToons [Sykora et al. 2009] that there is a tight relationship between Lumo and diffusion curves [Orzan et al. 2008], i.e., usage of Laplace equation for normal interpolation. It'd be nice to show a comparison with Lumo-like shading computed using harmonic energy (Figure 8 seems very close to what I have in mind) and denote that the proposed approach originate from a similar concept but can better preserve curvature lines. Besides Lumo I see also some resemblance to Surface Flows [Vergne et al. 2012]. It'd be interesting to provide a connection also with this method.
3) Quality of References	I think the following links to previous works are missing: 1) Alternative concepts for generating approximate 3D-like shading: Finch et al.: Freeform Vector Graphics with Controlled Thin-Plate Splines, SIGGRAPH Asia 2011. Vergne et al.: Surface Flows for Image-based Shading Design, SIGGRAPH 2012. Lopez-Moreno et al.: Depicting Stylized Materials with Vector Shade Trees, SIGGRAPH 2013. 2) Connection with Lumo and harmonic normal interpolation: Orzan et al.: Diffusion Curves: A Vector Representation for Smooth-Shaded Images, SIGGRAPH 2008. Sykora et al.: TexToons: Practical Texture Mapping for Hand-drawn Cartoon Animations, NPAR 2011. 3) Previous work on sketch-based 3D modelling as an alternative way to generate approximate shading: Igarashi et al. Teddy: A Sketching Interface for 3D Freeform Design, SIGGRAPH 1999. Karpenko & Hughes:. SmoothSketch: 3D Free-form Shapes from Complex Sketches, SIGGRAPH 2006. Wu et al.: ShapePalettes: Interactive Normal Transfer via Sketching, SIGGRAPH 2007. Nealen et al.: FiberMesh: Designing Freeform Surfaces with 3D Curves, SIGGRAPH 2007. Gingold et al.: Structured Annotations for 2D-to-3D Modeling, SIGGRAPH Asia 2009. Andre & Saito: Single-View Sketch Based Modeling, SBIM 2011.
4) Reproducibility	I think the paper is reproducible, however, requires deep knowledge of used mathematical concepts and some experience with tools able to solve mixed-integer optimization problems.
6) Rating	4.5
8) Explanation of Rating	I like this paper. The proposed method is clever and delivers impressive results. I have just few comments which might help to improve its quality. Besides better connection and comparison with previous work I'd like to see some more results (e.g., car, robot, or camera shown in [Shao et al. 2012]) and usage of more complex environment maps which can better reveal possible artifacts. It'd be also interesting to see an attempt to lower the computational overhead as well as to hear an opinion from real artists (e.g., via some informal user study). One of the limitations which I think should be discussed more thoroughly is the flattening artifact. I'm not sure if this is really caused only by erroneous orientation estimation at junctions. I think also the relative strength of local constraints plays an important role, e.g., locally boosting strength to the red curve which denotes shape of the chair's elbow might improve the results.

Reviewer #44:

Submission Information:

Question	Response
1) Description	This paper describes an interesting alternative approach to shading from concept sketches that does not require explicit 3d geometry. The results though are not convincing enough to make me think that this method is in any way superior to CrossShade, and the complexity of the optimization along with the relatively large run-time are further reasons that we would likely not see this implemented in any practical setting. This in itself is not a huge sleight, but it does harm the overall quality of the paper.
2) Clarity of Exposition	See below.
3) Quality of References	See below.
4) Reproducibility	See below.
6) Rating	3.2
8) Explanation of Rating	Solid introduction. I think i have an idea of what is going on, and the contributions are nicely framed and highlighted. The work is overall nicely motivated. The authors should perhaps mention that they are going for a (mostly) image-based approach to 3D lifting. Decent previous work, although with some redundancy, as many refs are already described in the intro. Figure 2 provides a very nice example of the method. What is p_ij? if this is described in [Kalberer 2007] then it needs to be mentioned here. It's explained later that the p_ij are the integers to be optimized. Maybe mention that earlier? This paper assumes too much prior knowledge in mixed integer optimization, which makes it harder to parse. maybe elaborate why each pixel has its own chart? I guess this is simple and easy to implement and fast enough, but I'd like to hear that from the authors. Figure 7 is a nice diagram that explains the integer optimization. But just to be clear, the only reason for integer optimization is that no two vector fields in a single chart can capture the flow fields due to umbilical and flat regions. I guess this is the price that needs to be paid, but it makes the optimization all but elegant. The move away from vector tools is sounding more and more contrived as I read more of the paper. The sketches shown can easily be converted to vector lines and I actually prefer the CrossShade normals.

Reviewer #77:

1) Description	In order to add believable shading a manually drawn 2D sketch, the 2D image is lifted to a 2.5D height field. This process works by defining a dense non orthogonal cross field in SCREEN SPACE, which follows and interpolates users strokes, and then finding the lifting which makes it orthogonal in OBJECT space. This process produces not only an approximate Height field, hence the normals, but also a local parameterization used for texture mapping. Many examples are shown.
2) Clarity of Exposition	Good enough. I feel like the paper *should* have included a 3D rendering of the height field. I feel like a demo *should* have been included. More details below, in "Explanation of Rating".
3) Quality of References	In not exactly an expert on sketch based modelling but a brief search did not reveal any important gap in reported literature. The literature of cross fields is covered sufficiently well, for this application. Maybe instead of the EG STAR Bommes et Al 2012, the more updated version (same authors, "Quad-Mesh Generation and Processing: A Survey", CG Forum, 2013) can be used.
4) Reproducibility	Yes, total. Makes you want to assign this as a project to some student. No passage is a mystery.
6) Rating	3.8
8) Explanation of Rating	In short, I just really liked the idea. I think it deserves the visibility of Siggraph. This is basically it. Below, I highlight a few detailed pros and cons of the paper and the technique, it their current form. The cons weaken the paper, (this is why I didn't score 4 or above), and fixing them would considerably strengthen it. But none of them outweighs the basic consideration above. In facts, the formalization, implementation recipe, presentation, and experiments are all already acceptable, if ameliorate. --------------- Pros: - Results do look nice, also compared to previous approaches. This is kind of subjective but inevitably so, and is also very relevant, considering this is all this kind of technique do is "prettyfing" an image (the output cannot be used as a 3D model, is not accurate enough to do anything else but reshading and a bit of retexturing, keeping the point of view fixed) - A plus, from the applicability point of view, is that simple raster images are used as input. This means that this technique is very flexible could be plugged into basically any sketch based system. - Cute how not only the shading but also texturing can be added over a sketch. --------------- Cons:

Submission Information:

Question	Response
	- [Not a con]: the paper might be criticized for being a simple idea with a limited scope for applicabilityl there are sketch based technique which are much more ambitious, recovering a full 3D model, not just enough 3D to attempt a better ligthing; but I like that it is self contained and (seemingly) just works. I don't consider this to be a con. - There is no executable prototype to test, given out as proof of concept. This omission is irritating and suspicious. I would have expected a running demo from this kind of work. I cannot see a reason why this has not been included. We reviewers (if not, any conference attendee) can very easily come up with usable input images of our own (exactly because the technique is claimed to be very "omnivore") and we could have seen for ourselves how well the technique performs of the average dataset. The technique looks simple enough but I don't have the time to re-implement it. - There is no validation of the basic assumption, which is that artist's stokes, and in particular lack of orthogonality of the implied field in image space, can be used in that way to guess the intended 3D shape (which means assuming that the artists actually follows that rule, and that they are skilled enough to stoke lines meeting at just the right angle). I understand this a validation not easy to get but an attempt should have been made: it would have made the paper stronger. [[A short digression about this: Here is an example of one such attempt to validate the assumption. Start from a 3D cad model. Show it to an artist, then ask him to draw it from memory, then use a synthetic image rendered form the original 3D model as a ground truth to measure how much the normals produced by the technique are accurate. Or, else, ask the artist to draw some object with some known a priori (for example, we know that two given surfaces should be orthogonal to each other in object space) and measure how much the a priori is satisfied by the recovered normals.]] - We never get to see, in the paper, how the recovered 2.5D fields really look like. A 10 pages long paper on such a self contained technique should not leave questions like this unanswered! A way to show them would be for example to change the point of view. I can understand the reason for this omission, but I don't agree with it. It is easy to guess is that the reason is that the 2.5D height field are never good enough to warrant a change of point of view; they are only good enough to recover reasonable normals. But this is only expected! I'm totally ok with this; but the paper should be honest and still show me how they look like and motivate why it is still ok. And, more imporntantly, show us *how* bad they are, exactly. For example, I'm left to wonder if the 2.5 height filed could be good enough to allow some simple global illumination technique, like ambient occlusion! - The math, notation, and formalization of the minimization systems used in section 6 are, at best, inelegant. For example, I am sure that there are ways to express the minimization problem of section 6 avoiding the need to subdivide the 2D image into regions. Here it would have been simpler and more elegant (and probably more efficient) to just define the local permutation needed to make any two neighboring cells "agree". Actually, the representation of the field is a bit naive and sub-optimal. My understanding is that a more suitable representation would result in better fields recovered from the same sparse inputs, as well as warrant a more elegant and more efficient minimization. IN SHORT, I didn't particularly like the way how the problem described in section 6 is tackled, compared to how similar problems are tackled in the literature. It looks naive and sub-optimal.

Submission Information: